



TUGAS AKHIR - SM141501

**ESTIMASI VARIABEL KEADAAN GERAK  
LONGITUDINAL PESAWAT TERBANG  
MENGUNAKAN METODE FUZZY KALMAN  
FILTER**

RESI ARUMIN SANI  
NRP 1212100039

Dosen Pembimbing:  
Prof. DR. Erna Apriliani, M.Si  
Prof. DR. Mohammad Isa Irawan, MT

JURUSAN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



FINAL PROJECT - SM141501

**VARIABLE STATE LONGITUDINAL MOTION  
AIRCRAFT ESTIMATION USING FUZZY  
KALMAN FILTER METHOD**

RESI ARUMIN SANI  
NRP 1212100039

Supervisors:

Prof. DR. Erna Apriliani, M.Si

Prof. DR. Mohammad Isa Irawan, MT

DEPARTMENT OF MATHEMATICS  
Faculty of Mathematics and Natural Sciences  
Sepuluh Nopember Institute of Technology  
Surabaya 2016


**LEMBAR PENGESAHAN**  
**ESTIMASI VARIABEL KEADAAN GERAK**  
**LONGITUDINAL PESAWAT TERBANG**  
**MENGGUNAKAN METODE FUZZY**  
**KALMAN FILTER**  
**VARIABLE STATE LONGITUDINAL**  
**MOTION AIRCRAFT ESTIMATION USING**  
**FUZZY KALMAN FILTER METHOD**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Sains  
pada  
Bidang Studi Matematika Terapan  
Program Studi S-1 Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:  
**RESI ARUMIN SANI**  
NRP. 1212100039

Menyetujui,  
Dosen Pembimbing II,

Dosen Pembimbing I,

  
**Prof. DR. M. Isa Irawan, MT**  
NIP. 19631225 198903 1 001

  
**Prof. DR. Erna Apriliani, M.Si**  
NIP. 19660414 199102 2 001

Mengetahui,  
Ketua Jurusan Matematika  
EMIPA ITS  
  
**Dr. Inam Mukhlash, S.Si, MT**  
NIP. 19700831 199403 1 003  
Surabaya, Juni 2016

# ESTIMASI VARIABEL KEADAAN GERAK LONGITUDINAL PESAWAT TERBANG MENGUNAKAN METODE FUZZY KALMAN FILTER

Nama Mahasiswa : Resi Arumin Sani  
NRP : 1212100039  
Jurusan : Matematika FMIPA-ITS  
Pembimbing : 1. Prof. DR. Erna Apriliani, M.Si  
2. Prof. DR. Mohammad Isa Irawan, MT

## Abstrak

*Pesawat terbang merupakan sarana transportasi udara yang memiliki enam derajat kebebasan gerak (DOF) yaitu sistem gerak yang dikontrol oleh aileron, elevator dan rudder. Gerak longitudinal pesawat terbang dikontrol oleh sistem elevator. Sistem pengendalian gerak pesawat terbang dengan mempertimbangkan adanya suatu noise, sehingga dibutuhkan estimator yang digunakan untuk mengestimasi gerak pesawat terbang. Estimasi dilakukan dengan metode Kalman Filter dan kombinasi Logika Fuzzy - Kalman Filter yang disebut Fuzzy Kalman Filter, serta optimal smoothing. Berdasarkan hasil penelitian, nilai root mean square error (RMSE) menunjukkan bahwa metode Fuzzy Kalman Filter relatif lebih kecil daripada metode Kalman Filter pada semua variabel gerak longitudinal pesawat terbang. Namun berdasarkan waktu komputasi, metode Kalman Filter lebih cepat dengan waktu 0.1628s dibandingkan Fuzzy Kalman Filter yang membutuhkan waktu 0.2802 s. Optimal smoothing Kalman Filter, estimasi variabel kecepatan translasi ke atas, kecepatan sudut pitch dan sudut pitch yang lebih halus. Optimal smoothing Fuzzy Kalman Filter, semua variabel dapat memberikan hasil yang lebih halus.*

***Kata-kunci:*** *Gerak Pesawat Terbang, Kalman Filter, Fuzzy Kalman Filter, Optimal Smoothing*

# VARIABLE STATE LONGITUDINAL MOTION AIRCRAFT ESTIMATION USING FUZZY KALMAN FILTER METHOD

Name : Resi Arumin Sani  
NRP : 1212100039  
Department : Mathematics FMIPA-ITS  
Supervisors : 1. Prof. DR. Erna Apriliani, M.Si  
2. Prof. DR. Mohammad Isa Irawan, MT

## Abstract

*The aircraft is the air transportation which has six degrees of freedom of motion (DOF) is system motion-controlled system of aileron, elevator and rudder. Longitudinal motion of the aircraft is controlled by a system of elevators. Motion control system aircraft with consider a noise, so it takes the estimator is used to estimate of aircraft motion. Estimates made by the method Kalman Filter and Fuzzy Logic combination of Kalman Filter called Fuzzy Kalman Filter, as well as the optimal smoothing. Based on the research results, the value root mean square error (RMSE) show that the method of Fuzzy Kalman Filter relatively smaller than the Kalman Filter all variable longitudinal motion of the aircraft. However based on the computation time, Kalman Filter more quick with a time of 0.1628 s compared Fuzzy Kalman Filter that takes 0.2802 s. Optimal smoothing Kalman Filter, estimation of variable translation velocity of upward, angular velocity of the pitch and pitch angular more subtle. In optimal smoothing Fuzzy Kalman Filter, all variables can provide a smoother result.*

**Keywords:** *Aircraft Motion, Kalman Filter, Fuzzy Kalman Filter, Optimal Smoothing*

## KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirobbil'aalamiin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan limpahan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

### **"ESTIMASI VARIABEL KEADAAN GERAK LONGITUDINAL PESAWAT TERBANG MENGUNAKAN METODE FUZZY KALMAN FILTER"**

sebagai salah satu syarat kelulusan Program Sarjana Jurusan Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Ketua Jurusan Matematika ITS yang telah memberikan motivasi selama perkuliahan hingga terselesaikannya Tugas Akhir ini.
2. Ibu Prof. DR. Erna Apriliani, M.Si dan Bapak Prof. DR. Mohammad Isa Irawan, MT selaku dosen pembimbing atas segala bimbingan, saran dan motivasinya kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.

3. Bapak Drs. Lukman Hanafi, M.Sc, Bapak Drs. Sentot Didik Surjanto, M.Si dan Ibu Dr. Dra. Mardlijah, MT selaku dosen penguji atas semua saran yang telah diberikan demi perbaikan Tugas Akhir ini.
4. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si selaku kepala program studi dan Bapak Drs. Iis Herisman, M.Sc selaku sekretaris program studi Tugas Akhir.
5. Bapak Drs. Komar Baihaqi, M.Si selaku dosen wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Jurusan Matematika FMIPA ITS.
6. Bapak dan Ibu dosen serta para staf Jurusan Matematika ITS yang tidak dapat penulis sebutkan satu-persatu.
7. Ibu dan Ayah, kedua orang tuaku tercinta, terima kasih atas segala do'anya dan kasih sayangnya yang selalu dicurahkan. Serta keluargaku, Kokoh, Kak Arum, Mas Hudi, om dan tante, bupo dan pakpo.
8. Nihaya Alivia Coraima Dewi yang menjadi penyemangatku. Terima kasih atas do'a dan motivasinya. Serta teman-teman angkatan 2012.

Penulis juga menyadari bahwa dalam Tugas Akhir ini masih terdapat kekurangan. Oleh sebab itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan Tugas Akhir ini. Akhirnya, penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi banyak pihak.

Surabaya, Juni 2016

Penulis



## DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	vi
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR SIMBOL	xxi
BAB I      PENDAHULUAN	1
1.1    Latar Belakang . . . . .	1
1.2    Rumusan Masalah . . . . .	3
1.3    Batasan Masalah . . . . .	3
1.4    Tujuan Penelitian . . . . .	4
1.5    Manfaat Penelitian . . . . .	4
1.6    Sistematika Penulisan . . . . .	4
BAB II     TINJAUAN PUSTAKA	7
2.1    Penelitian Terdahulu . . . . .	7
2.2    Model Dinamika Pesawat Terbang . . . . .	8
2.2.1    Komponen Gerak Pesawat Terbang . . .	8
2.2.2    Gerak Translasi . . . . .	9
2.2.3    Gerak Rotasi . . . . .	14
2.2.4    Linearisasi . . . . .	18

2.2.5	Persamaan Gerak Longitudinal Pesawat Terbang .....	28
2.3	Fuzzy Kalman Filter .....	30
2.3.1	Sistem Fuzzy .....	30
2.3.2	Fungsi Keanggotaan .....	32
2.3.3	Fuzzifikasi .....	34
2.3.4	Aturan Dasar Logika Fuzzy .....	35
2.3.5	Defuzzifikasi .....	35
2.3.6	Kalman Filter .....	36
2.3.7	Algoritma Fuzzy Kalman Filter .....	37
BAB III	METODE PENELITIAN .....	39
3.1	Studi Literatur .....	39
3.2	Mengkaji model gerak pesawat terbang, sistem Fuzzy dan Kalman Filter .....	39
3.3	Implementasi Metode .....	40
3.3.1	Implementasi Kalman Filter .....	40
3.3.2	Implementasi Fuzzy Kalman Filter ...	41
3.4	Simulasi dan Pembahasan .....	43
3.5	Penarikan Kesimpulan dan Saran .....	43
3.6	Pembuatan Laporan Tugas Akhir .....	44
BAB IV	HASIL DAN PEMBAHASAN .....	51
4.1	Persamaan Gerak Longitudinal Pesawat Terbang .....	51
4.1.1	Diskritisasi .....	53
4.1.2	Pembentukan Sistem Diskrit Stokastik .....	55
4.2	Implementasi Kalman Filter .....	57
4.3	Implementasi Fuzzy Kalman Filter .....	61
4.3.1	Fuzzifikasi .....	61
4.3.2	Aturan Dasar Logika Fuzzy .....	64
4.3.3	Algoritma Fuzzy Kalman Filter .....	64
4.3.4	Defuzzifikasi .....	77

4.4	Implementasi <i>Optimal Smoothing</i> Kalman Filter .....	78
4.5	Implementasi <i>Optimal Smoothing</i> Fuzzy Kalman Filter .....	80
4.6	Simulasi .....	83
4.6.1	Kasus 1 .....	86
4.6.2	Kasus 2 ( <i>Optimal Smoothing</i> Kalman Filter) .....	90
4.6.3	Kasus 3 ( <i>Optimal Smoothing</i> Fuzzy Kalman Filter) .....	93
BAB V	PENUTUP .....	99
5.1	Kesimpulan .....	99
5.2	Saran .....	100
DAFTAR PUSTAKA .....		101
LAMPIRAN .....		103
A	<i>Source Code</i> .....	105
B	Biodata Penulis .....	129

## DAFTAR GAMBAR

Gambar 2.1	Komponen arah gaya, momen dan kecepatan . . . . .	9
Gambar 2.2	Komponen gaya gravitasi . . . . .	12
Gambar 2.3	Komponen gaya gravitasi . . . . .	12
Gambar 2.4	Linear Naik . . . . .	33
Gambar 2.5	Linear Turun . . . . .	34
Gambar 3.1	Diagram Alur Penelitian . . . . .	45
Gambar 3.2	Diagram Alur Kalman Filter . . . . .	46
Gambar 3.3	Diagram Alur Fuzzy Kalman Filter . . . . .	47
Gambar 3.4	Diagram Alur Optimal Smoothing Kalman Filter . . . . .	48
Gambar 3.5	Diagram Alur Optimal Smoothing Fuzzy Kalman Filter . . . . .	49
Gambar 4.1	Diagram Blok Fuzzy untuk Model Gerak Longitudinal Pesawat Terbang . . . . .	63
Gambar 4.2	Bentuk tampilan GUI awal untuk simulasi estimasi gerak pesawat terbang . . . . .	85
Gambar 4.3	Hasil estimasi variabel gerak longitudinal pesawat terbang . . . . .	86
Gambar 4.4	Grafik estimasi kecepatan translasi ke depan ( $u$ ) . . . . .	87
Gambar 4.5	Grafik estimasi kecepatan translasi ke atas ( $w$ ) . . . . .	87
Gambar 4.6	Grafik estimasi kecepatan sudut <i>pitch</i> ( $q$ ) . . . . .	88
Gambar 4.7	Grafik estimasi sudut <i>pitch</i> ( $\theta$ ) . . . . .	88

Gambar 4.8	Hasil <i>optimal smoothing</i> Kalman Filter untuk variabel gerak longitudinal pesawat terbang .....	90
Gambar 4.9	Grafik <i>optimal smoothing</i> Kalman Filter pada kecepatan translasi ke depan ( $u$ ) .....	91
Gambar 4.10	Grafik <i>optimal smoothing</i> Kalman Filter pada kecepatan translasi ke atas ( $w$ ) .....	91
Gambar 4.11	Grafik <i>optimal smoothing</i> Kalman Filter pada kecepatan sudut <i>pitch</i> ( $q$ )..	92
Gambar 4.12	Grafik <i>optimal smoothing</i> Kalman Filter pada sudut <i>pitch</i> ( $\theta$ ) .....	92
Gambar 4.13	Hasil <i>optimal smoothing</i> Fuzzy Kalman Filter untuk variabel gerak longitudinal pesawat terbang .....	94
Gambar 4.14	Grafik <i>optimal smoothing</i> Fuzzy Kalman Filter pada kecepatan translasi ke depan ( $u$ ) .....	95
Gambar 4.15	Grafik <i>optimal smoothing</i> Fuzzy Kalman Filter pada kecepatan translasi ke atas ( $w$ ) .....	95
Gambar 4.16	Grafik <i>optimal smoothing</i> Fuzzy Kalman Filter pada kecepatan sudut <i>pitch</i> ( $q$ ) .....	96
Gambar 4.17	Grafik <i>optimal smoothing</i> Fuzzy Kalman Filter pada sudut <i>pitch</i> ( $\theta$ ) ...	96

## DAFTAR TABEL

Tabel 2.1	Komponen arah, gaya dan kecepatan . . . .	9
Tabel 4.1	Nilai parameter gerak longitudinal pesawat terbang . . . . .	57
Tabel 4.2	Nilai RMSE Kalman Filter dan Fuzzy Kalman Filter . . . . .	89
Tabel 4.3	Nilai Error Estimasi Kalman Filter dan Fuzzy Kalman Filter . . . . .	89
Tabel 4.4	Nilai RMSE Kalman Filter dan Optimal Smoothing . . . . .	93
Tabel 4.5	Nilai Error Estimasi Kalman Filter dan Optimal Smoothing . . . . .	93
Tabel 4.6	Nilai RMSE Fuzzy Kalman Filter dan Optimal Smoothing . . . . .	97
Tabel 4.7	Nilai Error Estimasi Fuzzy Kalman Filter dan Optimal Smoothing . . . . .	97

## Daftar Simbol

$P$	Kecepatan sudut sumbu <i>roll</i>
$Q$	Kecepatan sudut sumbu <i>pitch</i>
$R$	Kecepatan sudut sumbu <i>yaw</i>
$U$	Kecepatan translasi sumbu <i>roll</i>
$V$	Kecepatan translasi sumbu <i>pitch</i>
$W$	Kecepatan translasi sumbu <i>yaw</i>
$X$	Gaya sumbu <i>roll</i>
$Y$	Gaya sumbu <i>pitch</i>
$Z$	Gaya sumbu <i>yaw</i>
$L$	Momen torsi sumbu <i>roll</i>
$M$	Momen torsi sumbu <i>pitch</i>
$N$	Momen torsi sumbu <i>yaw</i>
$I_{xx}$	Momen inersia sumbu <i>roll</i>
$I_{yy}$	Momen inersia sumbu <i>pitch</i>
$I_{zz}$	Momen inersia sumbu <i>yaw</i>
$\Phi$	Perubahan sudut sumbu <i>roll</i>
$\Theta$	Perubahan sudut sumbu <i>pitch</i>
$\Psi$	Perubahan sudut sumbu <i>yaw</i>
$\Sigma F$	Resultan gaya dari pesawat terbang
$m$	Massa dari elemen pesawat terbang
$V_T$	Kecepatan translasi
$\omega$	Kecepatan rotasi ( <i>angular</i> )
$\dot{U}$	Percepatan translasi sumbu <i>roll</i>
$\dot{V}$	Percepatan translasi sumbu <i>pitch</i>
$\dot{W}$	Percepatan translasi sumbu <i>yaw</i>
$\Sigma F_x$	Resultan gaya sumbu <i>roll</i>
$\Sigma F_y$	Resultan gaya sumbu <i>pitch</i>
$\Sigma F_z$	Resultan gaya sumbu <i>yaw</i>
$g$	Gaya gravitasi pesawat terbang

$(F_x)_{gravity}$	Gaya gravitasi sumbu <i>roll</i>
$(F_y)_{gravity}$	Gaya gravitasi sumbu <i>pitch</i>
$(F_z)_{gravity}$	Gaya gravitasi sumbu <i>yaw</i>
$\Sigma M$	Resultan momen torsi dari pesawat terbang
$H$	Momen momentum
$I$	momen inersia
$\dot{P}$	Percepatan sudut sumbu <i>roll</i>
$\dot{Q}$	Percepatan sudut sumbu <i>pitch</i>
$\dot{R}$	Percepatan sudut sumbu <i>yaw</i>
$I_{xy}$	Momen inersia antara sumbu <i>roll</i> dan sumbu <i>pitch</i>
$I_{xz}$	Momen inersia antara sumbu <i>roll</i> dan sumbu <i>yaw</i>
$I_{yz}$	Momen inersia antara sumbu <i>pitch</i> dan sumbu <i>yaw</i>
$H_x$	Momen momentum sumbu <i>roll</i>
$H_y$	Momen momentum sumbu <i>pitch</i>
$H_z$	Momen momentum sumbu <i>yaw</i>
$M_x$	Momen torsi sumbu <i>roll</i>
$M_y$	Momen torsi sumbu <i>pitch</i>
$M_z$	Momen torsi sumbu <i>yaw</i>
$P_0$	Kecepatan sudut sumbu <i>roll</i> pada keadaan setimbang
$Q_0$	Kecepatan sudut sumbu <i>pitch</i> pada keadaan setimbang
$R_0$	Kecepatan sudut sumbu <i>yaw</i> pada keadaan setimbang
$U_0$	Kecepatan translasi sumbu <i>roll</i> pada keadaan setimbang
$V_0$	Kecepatan translasi sumbu <i>pitch</i> pada keadaan setimbang
$W_0$	Kecepatan translasi sumbu <i>yaw</i> pada keadaan setimbang
$X_0$	Gaya sumbu <i>roll</i> pada keadaan setimbang
$Y_0$	Gaya sumbu <i>pitch</i> pada keadaan setimbang
$Z_0$	Gaya sumbu <i>yaw</i> pada keadaan setimbang
$L_0$	Momen torsi sumbu <i>roll</i> pada keadaan setimbang
$M_0$	Momen torsi sumbu <i>pitch</i> pada keadaan setimbang
$N_0$	Momen torsi sumbu <i>yaw</i> pada keadaan setimbang
$\Phi_0$	Perubahan sudut sumbu <i>roll</i> pada keadaan setimbang
$\Theta_0$	Perubahan sudut sumbu <i>pitch</i> pada keadaan setimbang
$\Psi_0$	Perubahan sudut sumbu <i>yaw</i> pada keadaan setimbang
$\Delta P, p$	Kecepatan sudut sumbu <i>roll</i> pada keadaan gangguan



$\Delta Q, q$	Kecepatan sudut sumbu <i>pitch</i> pada keadaan gangguan
$\Delta R, r$	Kecepatan sudut sumbu <i>yaw</i> pada keadaan gangguan
$\Delta U, u$	Kecepatan translasi sumbu <i>roll</i> pada keadaan gangguan
$\Delta V, v$	Kecepatan translasi sumbu <i>pitch</i> pada keadaan gangguan
$\Delta W, w$	Kecepatan translasi sumbu <i>yaw</i> pada keadaan gangguan
$\Delta X, x$	Gaya sumbu <i>roll</i> pada keadaan gangguan
$\Delta Y, y$	Gaya sumbu <i>pitch</i> pada keadaan gangguan
$\Delta Z, z$	Gaya sumbu <i>yaw</i> pada keadaan gangguan
$\Delta L, l$	Momen torsi sumbu <i>roll</i> pada keadaan gangguan
$\Delta M, m$	Momen torsi sumbu <i>pitch</i> pada keadaan gangguan
$\Delta N, n$	Momen torsi sumbu <i>yaw</i> pada keadaan gangguan
$\Delta \Phi, \phi$	Perubahan sudut sumbu <i>roll</i> pada keadaan gangguan
$\Delta \Theta, \theta$	Perubahan sudut sumbu <i>pitch</i> pada keadaan gangguan
$\Delta \Psi, \psi$	Perubahan sudut sumbu <i>yaw</i> pada keadaan gangguan
$\delta_a$	Defleksi <i>aileron</i>
$\delta_e$	Defleksi <i>elevator</i>
$\delta_r$	Defleksi <i>rudder</i>
$\delta_T$	Defleksi <i>throttle</i>
$X_u$	Gaya sumbu <i>roll</i> terhadap $u$ per massa
$X_w$	Gaya sumbu <i>roll</i> terhadap $w$ per massa
$X_{\delta_e}$	Gaya sumbu <i>roll</i> terhadap $\delta_e$ per massa
$X_{\delta_T}$	Gaya sumbu <i>roll</i> terhadap $\delta_T$ per massa
$Z_u$	Gaya sumbu <i>yaw</i> terhadap $u$ per massa
$Z_w$	Gaya sumbu <i>yaw</i> terhadap $w$ per massa
$Z_{\dot{w}}$	Gaya sumbu <i>yaw</i> terhadap $\dot{w}$ per massa
$Z_q$	Gaya sumbu <i>yaw</i> terhadap $q$ per massa
$Z_{\delta_e}$	Gaya sumbu <i>yaw</i> terhadap $\delta_e$ per massa
$Z_{\delta_T}$	Gaya sumbu <i>yaw</i> terhadap $\delta_T$ per massa
$M_u$	Momen torsi sumbu <i>pitch</i> terhadap $u$ per momen inersia
$M_w$	Momen torsi sumbu <i>pitch</i> terhadap $w$ per momen inersia
$M_{\dot{w}}$	Momen torsi sumbu <i>pitch</i> terhadap $\dot{w}$ per momen inersia
$M_q$	Momen torsi sumbu <i>pitch</i> terhadap $q$ per momen inersia
$M_{\delta_e}$	Momen torsi sumbu <i>pitch</i> terhadap $\delta_e$ per momen inersia
$M_{\delta_T}$	Momen torsi sumbu <i>pitch</i> terhadap $\delta_T$ per momen inersia

$x_{k+1}$	Variabel keadaan pada waktu $k + 1$
$x_k$	Variabel keadaan pada waktu $k$
$A_k$	Matriks koefisien
$B_k$	Matriks koefisien untuk input sistem
$u_k$	Variabel input sistem
$G_k$	Matriks koefisien noise sistem
$\xi_k$	Noise sistem
$z_k$	Vektor pengukuran pada waktu ke- $k$
$H_k$	Matriks koefisien pengukuran
$\eta_k$	Noise pengukuran
$P_k$	Matriks kovarian error pada waktu $k$
$Q_k$	Matriks kovarian error noise sistem pada waktu $k$
$R_k$	Matriks kovarian error noise pengukuran pada waktu $k$
$x_{i,k+1}$	Variabel keadaan pada waktu $k + 1$ untuk aturan ke- $i$
$\hat{x}_{\bar{k}+1}$	Estimasi variabel keadaan pada waktu $k + 1$ sebelum pengukuran
$\hat{x}_{i,\bar{k}+1}$	Estimasi variabel keadaan pada waktu $k + 1$ untuk aturan ke- $i$ sebelum pengukuran
$\hat{x}_{k+1}$	Estimasi variabel keadaan pada waktu $k + 1$ setelah pengukuran
$\hat{x}_{i,k+1}$	Estimasi variabel keadaan pada waktu $k + 1$ untuk aturan ke- $i$ setelah pengukuran
$P_{\bar{k}+1}$	Matriks kovarian error pada waktu $k + 1$ sebelum pengukuran
$P_{i,\bar{k}+1}$	Matriks kovarian error pada waktu $k + 1$ untuk aturan ke- $i$ sebelum pengukuran
$P_{k+1}$	Matriks kovarian error pada waktu $k + 1$ setelah pengukuran
$P_{i,k+1}$	Matriks kovarian error pada waktu $k + 1$ untuk aturan ke- $i$ setelah pengukuran
$K_{k+1}$	Kalman Gain pada waktu $k + 1$
$K_{i,k+1}$	Kalman Gain pada waktu $k + 1$ untuk aturan ke- $i$
$\mu_A^i$	Fungsi keanggotaan dari A
$\rho^i$	Himpunan Fuzzy untuk rumus bobot rata-rata

# BAB I

## PENDAHULUAN

Pada bab ini dijelaskan mengenai hal-hal yang menjadi latar belakang munculnya permasalahan yang dibahas dalam Tugas Akhir ini. Permasalahan-permasalahan tersebut disusun ke dalam suatu rumusan masalah. Selanjutnya dijabarkan juga batasan masalah untuk mendapatkan tujuan yang diinginkan serta manfaat yang dapat diperoleh dari Tugas Akhir ini. Adapun sistematika penulisan Tugas Akhir diuraikan pada bagian akhir bab ini.

### 1.1 Latar Belakang

Pesawat udara merupakan sarana transportasi udara yang diperlukan bagi khalayak umum. Pesawat udara dibagi menjadi dua yaitu pesawat bersayap tetap dan pesawat bersayap putar. Pesawat telah dirancang dan dibangun sesuai kebutuhan dengan perhitungan yang akurat. Pesawat membutuhkan tingkat pengendalian yang optimal untuk dapat terbang stabil. Pesawat terbang memiliki enam derajat kebebasan gerak (*Six Degree of Freedom*) yang mana masing-masing dikontrol oleh sistem *aileron*, sistem *elevator* dan sistem *rudder*. Berdasarkan arah, gerak terbang pesawat terdapat dua yaitu gerak longitudinal dan gerak lateral. Dengan adanya faktor-faktor internal dan eksternal dari pesawat yang dapat menghambat kestabilan pesawat saat terbang di udara. Faktor-faktor internal yang meliputi sistem kontrol gerak, roda dan pada mesinnya. Faktor-faktor eksternal meliputi kondisi cuaca buruk seperti badai angin, awan, hujan disertai petir.

Sistem pengendalian pesawat yang telah dirancang sesuai kebutuhan dengan tingkat pengukuran yang akurat akan terdapat suatu noise. Noise ukurannya sangat kecil. Noise tersebut dapat terjadi pada noise sistem dan noise pengukuran. Meskipun ukurannya sangat kecil, noise-noise tersebut dapat menghambat kinerja dari sistem pengendalian gerak pesawat. Noise tersebut harus dihilangkan untuk mengatasi masalah kinerja dari sistem pengendalian gerak pesawat. Dengan mengurangi noise pada sistem dan pengukuran tersebut diperlukan adanya suatu pendekatan yang lebih akurat dari sebelumnya. Suatu pendekatan yang dilakukan yaitu berupa adanya estimator untuk mengetahui tingkat noise tersebut. Estimator digunakan untuk memprediksi variabel-variabel kontrol gerak pesawat dengan adanya noise tersebut.

Berdasarkan hasil penelitian yang dilakukan oleh Abhijit G. Kallapur pada tahun 2007 tentang "UAV Linear and Nonlinear Estimation Using Extended Kalman Filter" [1]. Tingkat akurasi estimasi masih cenderung kurang optimal. Berdasarkan hasil penelitian yang dilakukan oleh Hasnawati Mahmuri pada tahun 2011 tentang "Estimasi Perkembangan Sel Kanker Menggunakan Fuzzy Kalman Filter" [4]. Metode kombinasi Fuzzy Kalman Filter memberikan hasil estimasi lebih akurat dibandingkan dengan metode Kalman Filter biasa.

Fuzzy Kalman Filter merupakan suatu metode fusi yang berasal dari Logika Fuzzy dan Kalman Filter. Logika Fuzzy merupakan metode penalaran yang dapat menangani masalah ketidakpastian pada dunia nyata. Sedangkan Kalman Filter merupakan metode estimasi variabel keadaan sistem linear dinamik. Metode Fuzzy Kalman Filter ini telah diteliti oleh Lee Ryeok Han pada thesisnya berjudul "A Fuzzy-Kalman Filtering Strategy for State Estimation" pada tahun 2004

[9]. Berdasarkan hasil percobaan yang dilakukan Lee Ryeok Han menyimpulkan bahwa metode Fuzzy Kalman Filter suatu kombinasi yang tepat pada penanganan masalah yang tidak pasti dan lebih akurat dibandingkan Kalman Filter.

Dalam Tugas Akhir ini penulis bermaksud membahas topik "Estimasi Variabel Keadaan Gerak Pesawat Terbang Menggunakan Metode Fuzzy Kalman Filter". Pada penelitian ini juga dilakukan *optimal smoothing* pada Kalman Filter dan Fuzzy Kalman Filter. Penelitian ini bermaksud untuk mengestimasi variabel keadaan gerak untuk mendapatkan tingkat kesalahan estimasi yang sangat kecil.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disajikan diatas, penulis menuliskan beberapa permasalahan-permasalahan yang akan dibahas dalam penelitian Tugas Akhir ini sebagai berikut :

1. Bagaimana cara mengkaji model dinamika gerak pesawat terbang?
2. Bagaimana hasil estimasi algoritma Kalman Filter dengan algoritma Fuzzy Kalman Filter?
3. Bagaimana hasil dari *Optimal Smoothing* terhadap kedua metode tersebut?

## 1.3 Batasan Masalah

Dalam Tugas Akhir ini, penulis membatasi permasalahan sebagai berikut :

1. Model pesawat yang diteliti merupakan pesawat bersayap tetap (*fixed wing*).
2. Sistem model kontrol gerak pesawat yang diteliti adalah pada gerakan longitudinal.

3. Software yang digunakan untuk simulasi adalah MATLAB.

#### **1.4 Tujuan Penelitian**

Tujuan dari penelitian Tugas Akhir ini adalah merancang estimasi pada variabel keadaan sistem kontrol pesawat pada gerak longitudinal dengan menggunakan metode Fuzzy Kalman Filter. Dengan perancangan estimasi ini agar dapat memberikan tingkat kesalahan estimasi yang kecil.

#### **1.5 Manfaat Penelitian**

Dari penelitian Tugas Akhir ini, penulis mengharapkan agar Tugas Akhir ini dapat bermanfaat bagi berbagai kalangan sebagai berikut :

1. Memperluas masalah-masalah yang dapat diterapkan dengan metode Fuzzy Kalman Filter.
2. Menambah wawasan dan memberi gambaran tentang estimasi dengan menggunakan algoritma Fuzzy Kalman Filter.
3. Sebagai bahan pertimbangan dalam estimasi sistem kontrol gerak pesawat untuk merancang sebuah pesawat.

#### **1.6 Sistematika Penulisan**

Penulisan Tugas Akhir ini disusun dalam lima bab, yaitu:

##### **1. BAB I PENDAHULUAN**

Pada bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

##### **2. BAB II TINJAUAN PUSTAKA**

Pada Bab ini berisi tentang penelitian terdahulu,

model dinamika gerak longitudinal pesawat terbang dan Metode Kalman Filter serta Fuzzy Kalman Filter.

### 3. BAB III METODE PENELITIAN

Pada bab ini dijelaskan tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir. Tahapan-tahapan tersebut antara lain studi literatur, mengkaji model gerak pesawat terbang, Kalman Filter dan Fuzzy Kalman Filter. Selanjutnya dilakukan implementasi metode Kalman Filter dan Fuzzy Kalman Filter. Tahap selanjutnya dilakukan simulasi dan analisis hasil. Tahap terakhir adalah melakukan penarikan kesimpulan berdasarkan hasil analisis serta saran.

### 4. BAB IV HASIL DAN PEMBAHASAN

Pada Bab ini dibahas mengenai penerapan model dinamika gerak longitudinal pesawat terbang dengan metode Kalman Filter dan Fuzzy Kalman Filter. Dan selanjutnya akan diperoleh hasil estimasi dari Kalman Filter dan Fuzzy Kalman Filter.

### 5. BAB V PENUTUP

Pada bab ini berisi mengenai kesimpulan akhir yang diperoleh dari Tugas Akhir serta saran untuk pengembangan penelitian selanjutnya.

*"Halaman ini sengaja dikosongkan."*



## BAB II

### TINJAUAN PUSTAKA

Pada bab ini diuraikan mengenai hasil dari penelitian-penelitian sebelumnya yang terkait dengan permasalahan dalam Tugas Akhir ini. Selain itu juga diuraikan mengenai model dinamika gerak longitudinal pesawat terbang serta algoritma Kalman Filter dan algoritma Fuzzy Kalman Filter.

#### 2.1 Penelitian Terdahulu

Dalam Tugas Akhir ini penulis merujuk pada beberapa penelitian-penelitian sebelumnya yang sesuai dengan topik yang diambil. Salah satu penelitian yang digunakan adalah jurnal yang ditulis oleh Abhijit G. Kallapur pada tahun 2007 yang berjudul "*UAV Linear and Nonlinear Estimation Using Extended Kalman Filter*". Pada penelitian tersebut menyatakan bahwa estimasi dari model dinamika UAV yang terdiri dari satu model linear dan dua model nonlinear telah menunjukkan kecenderungan hasil yang sama. Nilai *Root Mean Square Error* (RMSE) untuk ketiga model dinamik UAV hampir identik. Hilangnya tingkat akurasi dalam estimasi model dinamika UAV disebabkan mengabaikan model linear [1]. Pada penelitian lain yang ditulis oleh Agus Sukandi pada tahun 2010 yang berjudul "*Pengendalian Gerak Longitudinal Pesawat Terbang dengan Metode Decoupling*". Pada penelitian tersebut diperoleh model gerak longitudinal pesawat terbang dengan nilai parameter-parameter yang diketahui [5].

Pada penelitian yang lain yaitu thesis yang ditulis oleh Hasnawati Mahmuri pada tahun 2011 yang berjudul

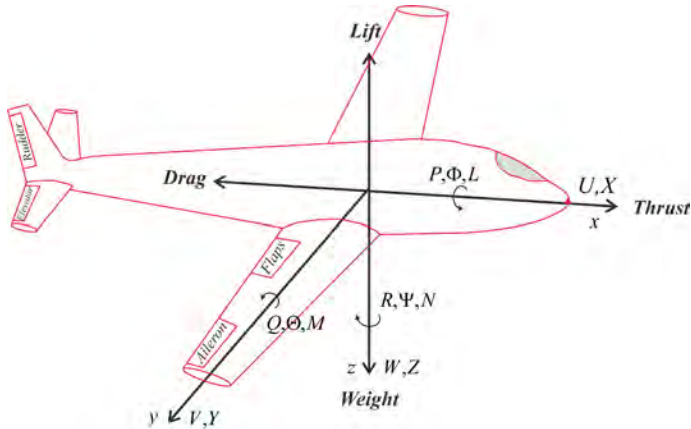
"Estimasi Perkembangan Sel Kanker Menggunakan Fuzzy Kalman Filter". Pada penelitian tersebut menyatakan bahwa error estimasi Fuzzy Kalman Filter lebih kecil daripada error Kalman Filter sehingga algoritma Fuzzy Kalman Filter dapat digunakan untuk mengestimasi variabel keadaan yang lebih akurat [4]. Berdasarkan penelitian-penelitian tersebut, pada Tugas Akhir ini akan dilakukan estimasi variabel gerak pesawat terbang menggunakan metode Fuzzy Kalman Filter.

## 2.2 Model Dinamika Pesawat Terbang

### 2.2.1 Komponen Gerak Pesawat Terbang

Model dinamika gerak pesawat terbagi menjadi dua gerakan yaitu gerak longitudinal dan gerak lateral. Pesawat memiliki enam derajat kebebasan (*Six Degree of Freedom*) dalam pergerakannya. Untuk dapat bergerak dalam enam derajat kebebasan, pesawat memiliki sistem kontrol yaitu *aileron*, *elevator* dan *rudder*. Pada gerak longitudinal pesawat dikendalikan oleh sayap elevator. Sayap elevator merupakan sistem kontrol gerak naik turunnya sayap kiri atau kanan pesawat (*pitching*). Pada gerak lateral pesawat dikendalikan oleh sayap *aileron* dan sayap *rudder*. Sayap aileron merupakan sistem kontrol gerak roll sayap kiri atau kanan pesawat (*rolling*). Sayap rudder merupakan sistem kontrol gerak belok ke kiri atau kanan (*yawing*). Namun, selain tiga sistem kontrol tersebut terdapat *throttle* yang dapat memberikan pengaruh dari perubahan sudut elevasi dari sistem kontrol gerak pesawat. *Throttle* berfungsi untuk mengatur *thrust*/gaya dorong dari mesin pesawat, sehingga akan berpengaruh pada kecepatan pesawat [6].

Pesawat mampu bergerak dalam tiga sumbu  $(x, y, z)$  yaitu *rolling*, *pitching* dan *yawing*. Komponen arah gaya, momen, kecepatan translasi dan rotasi (*angular*) pada koordinat  $(x, y, z)$  dilihat pada Gambar (2.1) [5].



Gambar 2.1: Komponen arah gaya, momen dan kecepatan

Tabel 2.1: Komponen arah, gaya dan kecepatan

Komponen	Roll	Pitch	Yaw
Kecepatan Sudut	$P$	$Q$	$R$
Kecepatan Translasi	$U$	$V$	$W$
Gaya	$X$	$Y$	$Z$
Momen Torsi	$L$	$M$	$N$
Momen Inersia	$I_{xx}$	$I_{yy}$	$I_{zz}$
Perubahan Sudut	$\Phi$	$\Theta$	$\Psi$

## 2.2.2 Gerak Translasi

- Gaya yang bekerja karena gaya dorong mesin pesawat terbang

Gaya yang bekerja karena gaya dorong mesin pesawat terbang dipengaruhi oleh kecepatan translasi dan kecepatan rotasi. Maka persamaan gaya dorong mesin [5] sebagai berikut:

Berdasarkan prinsip Hukum Newton II, yaitu

$$\begin{aligned}\sum F &= ma \\ \sum F &= m \left\{ \frac{d}{dt} V_T \right\} \\ \sum F &= m \left\{ \frac{d}{dt} V_T + (\omega \times V_T) \right\} \quad (2.1)\end{aligned}$$

dimana :

$\sum F$  : resultan gaya dari pesawat terbang  $[N]$   
 $m$  : massa dari elemen pesawat terbang  $[kg]$   
 $V_T$  : kecepatan translasi  $[m/s]$   
 $\omega$  : kecepatan rotasi (*angular*)  $[rad/s]$

Dengan diketahui kecepatan translasi dan rotasi masing-masing terhadap sumbunya, yaitu

$$\begin{aligned}V_T &= U\hat{i} + V\hat{j} + W\hat{k} \\ \omega &= P\hat{i} + Q\hat{j} + R\hat{k}\end{aligned}$$

sehingga diperoleh

$$\frac{d}{dt} V_T = \frac{d}{dt} \{U\hat{i} + V\hat{j} + W\hat{k}\} = \dot{U}\hat{i} + \dot{V}\hat{j} + \dot{W}\hat{k} \quad (2.2)$$

Dengan

$$\begin{aligned}\omega \times V_T &= (P\hat{i} + Q\hat{j} + R\hat{k}) \times (U\hat{i} + V\hat{j} + W\hat{k}) \\ \omega \times V_T &= PU(\hat{i} \times \hat{i}) + PV(\hat{i} \times \hat{j}) + PW(\hat{i} \times \hat{k}) + \\ &\quad QU(\hat{j} \times \hat{i}) + QV(\hat{j} \times \hat{j}) + QW(\hat{j} \times \hat{k}) + \\ &\quad RU(\hat{k} \times \hat{i}) + RV(\hat{k} \times \hat{j}) + RW(\hat{k} \times \hat{k})\end{aligned}$$

$$\begin{aligned}\omega \times V_T &= PV\hat{k} + PW(-\hat{j}) + QU(-\hat{k}) + QW\hat{i} + \\ &\quad RU\hat{j} + RV(-\hat{i})\end{aligned}$$

$$\omega \times V_T = (QW - RV)\hat{i} + (RU - PW)\hat{j} + (PV - QU)\hat{k}$$

maka,

$$\begin{aligned}\sum F &= m \left\{ \frac{d}{dt} V_T + (\omega \times V_T) \right\} \\ &= m \left\{ (\dot{U}\hat{i} + \dot{V}\hat{j} + \dot{W}\hat{k}) + \left( (QW - RV)\hat{i} + \right. \right. \\ &\quad \left. \left. (RU - PW)\hat{j} + (PV - QU)\hat{k} \right) \right\} \\ &= m \left\{ (\dot{U} + QW - RV)\hat{i} + (\dot{V} + RU - PW)\hat{j} + \right. \\ &\quad \left. (\dot{W} + PV - QU)\hat{k} \right\} \\ &= m (\dot{U} + QW - RV)\hat{i} + m (\dot{V} + RU - PW)\hat{j} + \\ &\quad m (\dot{W} + PV - QU)\hat{k}\end{aligned}$$

karena

$$\sum F = \sum F_x \hat{i} + \sum F_y \hat{j} + \sum F_z \hat{k}$$

sehingga diperoleh [5]:

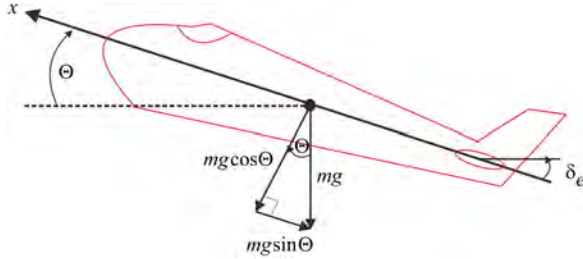
$$\sum F_x = m (\dot{U} + QW - RV) \quad (2.3)$$

$$\sum F_y = m (\dot{V} + RU - PW) \quad (2.4)$$

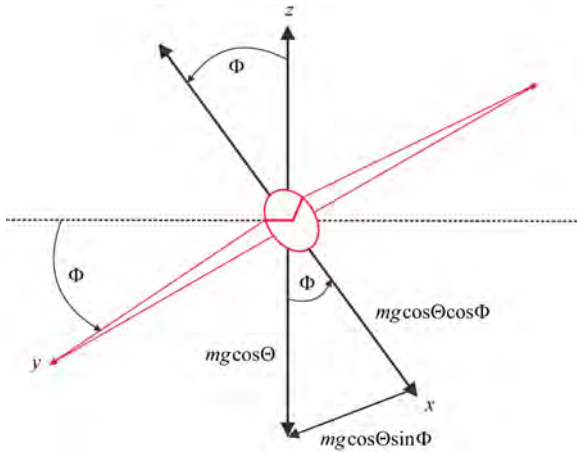
$$\sum F_z = m (\dot{W} + PV - QU) \quad (2.5)$$

- Gaya yang bekerja karena gravitasi bumi

Gaya yang bekerja karena pengaruh gaya gravitasi bumi dapat digambarkan pada Gambar (2.2) dan Gambar (2.3), sebagai berikut :



Gambar 2.2: Komponen gaya gravitasi



Gambar 2.3: Komponen gaya gravitasi

Persamaan gaya gravitasi pada masing-masing komponen diperoleh sebagai berikut :

$$(F_x)_{gravity} = -mg \sin \Theta \quad (2.6)$$

$$(F_y)_{gravity} = mg \cos \Theta \sin \Phi \quad (2.7)$$

$$(F_z)_{gravity} = mg \cos \Theta \cos \Phi \quad (2.8)$$

Dari persamaan 2.3 - 2.5 dan 2.6 - 2.8 diperoleh gaya total masing-masing sumbu :

$$\begin{aligned}\sum F_x &= F_x + (F_x)_{gravity} = m \left( \dot{U} + QW - RV \right) \\ \sum F_y &= F_y + (F_y)_{gravity} = m \left( \dot{V} + RU - PW \right) \\ \sum F_z &= F_z + (F_z)_{gravity} = m \left( \dot{W} + PV - QU \right)\end{aligned}$$

sehingga diperoleh,

$$F_x - mg \sin \Theta = m \left( \dot{U} + QW - RV \right) \quad (2.9)$$

$$F_y + mg \cos \Theta \sin \Phi = m \left( \dot{V} + RU - PW \right) \quad (2.10)$$

$$F_z + mg \cos \Theta \cos \Phi = m \left( \dot{W} + PV - QU \right) \quad (2.11)$$

Karena  $F_x = X, F_y = Y, F_z = Z$ , sehingga persamaan gaya yang bekerja pada sumbu  $(x, y, z)$  menjadi [5]:

$$\begin{aligned}X - mg \sin \Theta &= m \left( \dot{U} + QW - RV \right) \\ Y + mg \cos \Theta \sin \Phi &= m \left( \dot{V} + RU - PW \right) \\ Z + mg \cos \Theta \cos \Phi &= m \left( \dot{W} + PV - QU \right)\end{aligned}$$

Dan dapat ditulis menjadi

$$X = m \left( \dot{U} + QW - RV + g \sin \Theta \right) \quad (2.12)$$

$$Y = m \left( \dot{V} + RU - PW - g \cos \Theta \sin \Phi \right) \quad (2.13)$$

$$Z = m \left( \dot{W} + PV - QU + g \cos \Theta \cos \Phi \right) \quad (2.14)$$

### 2.2.3 Gerak Rotasi

Pada gerak rotasi pesawat terbang menggunakan persamaan momen torsi. Torsi pada *center of gravity* pesawat terbang, sebagai berikut :

$$\begin{aligned}\sum M &= \frac{d}{dt}H + (\omega \times H) \\ \sum M &= \frac{d}{dt}(I\omega) + (\omega \times H) \\ \sum M &= I\frac{d\omega}{dt} + (\omega \times H)\end{aligned}\tag{2.15}$$

dimana :

$M$  : momen torsi  $[kgm^2rad/s^2]$   
 $H$  : momen momentum  $[kgm^2rad/s]$   
 $I$  : momen inersia  $[kgm^2]$

Dengan mendefinisikan matriks momen inersia, sebagai berikut [10]:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}\tag{2.16}$$

Maka untuk

$$\frac{d\omega}{dt} = \frac{d}{dt} \{P\hat{i} + Q\hat{j} + R\hat{k}\} = \dot{P}\hat{i} + \dot{Q}\hat{j} + \dot{R}\hat{k}$$

untuk

$$\begin{aligned}H &= I\omega \\ H &= \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \\ H &= \begin{bmatrix} I_{xx}P - I_{xy}Q - I_{xz}R \\ -I_{xy}P + I_{yy}Q - I_{yz}R \\ -I_{xz}P - I_{yz}Q + I_{zz}R \end{bmatrix}\end{aligned}$$



Sehingga diperoleh

$$H_x = I_{xx}P - I_{xy}Q - I_{xz}R \quad (2.17)$$

$$H_y = -I_{xy}P + I_{yy}Q - I_{yz}R \quad (2.18)$$

$$H_z = -I_{xz}P - I_{yz}Q + I_{zz}R \quad (2.19)$$

dan

$$I \frac{d\omega}{dt} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix}$$

$$I \frac{d\omega}{dt} = \begin{bmatrix} I_{xx}\dot{P} - I_{xy}\dot{Q} - I_{xz}\dot{R} \\ -I_{xy}\dot{P} + I_{yy}\dot{Q} - I_{yz}\dot{R} \\ -I_{xz}\dot{P} - I_{yz}\dot{Q} + I_{zz}\dot{R} \end{bmatrix}$$

Karena secara umum, pesawat terbang simetri terhadap sumbu  $xz$  dan berakibat [10],

$$I_{xy} = I_{yz} = 0$$

maka persamaan 2.16 - 2.18 menjadi :

$$H_x = I_{xx}P - I_{xz}R \quad (2.20)$$

$$H_y = I_{yy}Q \quad (2.21)$$

$$H_z = -I_{xz}P + I_{zz}R \quad (2.22)$$

dan juga,

$$I \frac{d\omega}{dt} = \begin{bmatrix} I_{xx}\dot{P} - I_{xz}\dot{R} \\ I_{yy}\dot{Q} \\ -I_{xz}\dot{P} + I_{zz}\dot{R} \end{bmatrix} \quad (2.23)$$

Dan untuk

$$\begin{aligned}
\omega \times H &= (P\hat{i} + Q\hat{j} + R\hat{k}) \times (H_x\hat{i} + H_y\hat{j} + H_z\hat{k}) \\
&= PH_x(\hat{i} \times \hat{i}) + PH_y(\hat{i} \times \hat{j}) + PH_z(\hat{i} \times \hat{k}) + \\
&\quad QH_x(\hat{j} \times \hat{i}) + QH_y(\hat{j} \times \hat{j}) + QH_z(\hat{j} \times \hat{k}) + \\
&\quad RH_x(\hat{k} \times \hat{i}) + RH_y(\hat{k} \times \hat{j}) + RH_z(\hat{k} \times \hat{k}) \\
&= PH_y\hat{k} + PH_z(-\hat{j}) + QH_x(-\hat{k}) + QH_z\hat{i} + \\
&\quad RH_x\hat{j} + RH_y(-\hat{i})
\end{aligned}$$

sehingga

$$\begin{aligned}
\omega \times H &= (QH_z - RH_y)\hat{i} + (RH_x - PH_z)\hat{j} + \\
&\quad (PH_y - QH_x)\hat{k}
\end{aligned} \tag{2.24}$$

Substitusi persamaan 2.20 - 2.22 ke persamaan 2.24, sehingga diperoleh

$$\begin{aligned}
\omega \times H &= (Q\{-I_{xz}P + I_{zz}R\} - R\{I_{yy}Q\})\hat{i} + \\
&\quad (R\{I_{xx}P - I_{xz}R\} - P\{-I_{xz}P + I_{zz}R\})\hat{j} + \\
&\quad (P\{I_{yy}Q\} - Q\{I_{xx}P - I_{xz}R\})\hat{k} \\
\omega \times H &= (-I_{xz}PQ + I_{zz}RQ - I_{yy}QR)\hat{i} + \\
&\quad (I_{xx}PR - I_{xz}R^2 + I_{xz}P^2 - I_{zz}PR)\hat{j} + \\
&\quad (I_{yy}PQ - I_{xx}PQ + I_{xz}QR)\hat{k}
\end{aligned} \tag{2.25}$$

Sehingga untuk mendapatkan momen torsi dengan substitusi persamaan 2.23 dan persamaan 2.25 ke persamaan 2.15

menjadi :

$$\begin{aligned}
\sum M &= \left\{ (I_{xx}\dot{P} - I_{xz}\dot{R})\hat{i} + I_{yy}\dot{Q}\hat{j} + (-I_{xz}\dot{P} + I_{zz}\dot{R})\hat{k} \right\} + \\
&\quad \left\{ (-I_{xz}PQ + I_{zz}RQ - I_{yy}QR)\hat{i} + \right. \\
&\quad (I_{xx}PR - I_{xz}R^2 + I_{xz}P^2 - I_{zz}PR)\hat{j} + \\
&\quad \left. (I_{yy}PQ - I_{xx}PQ + I_{xz}QR)\hat{k} \right\} \\
\sum M &= (I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR)\hat{i} + \\
&\quad (I_{yy}\dot{Q} + I_{xz}(P^2 - R^2) + (I_{xx} - I_{zz})PR)\hat{j} + \\
&\quad (I_{zz}\dot{R} - I_{xz}(\dot{P} - QR) + (I_{yy} - I_{xx})PQ)\hat{k} \quad (2.26)
\end{aligned}$$

Maka dapat ditulis dalam bentuk, sebagai berikut :

$$\sum M = M_x\hat{i} + M_y\hat{j} + M_z\hat{k}$$

Sehingga nilai masing-masing sumbunya, yaitu

$$M_x = I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR \quad (2.27)$$

$$M_y = I_{yy}\dot{Q} + I_{xz}(P^2 - R^2) + (I_{xx} - I_{zz})PR \quad (2.28)$$

$$M_z = I_{zz}\dot{R} - I_{xz}(\dot{P} - QR) + (I_{yy} - I_{xx})PQ \quad (2.29)$$

Karena komponen momen torsi pada masing-masing sumbu *roll*, *pitch*, *yaw* yaitu

$$M_x = L, M_y = M, M_z = N$$

Sehingga momen torsi *roll*, *pitch*, *yaw* pada persamaan 2.27 - 2.29 menjadi

$$L = I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR \quad (2.30)$$

$$M = I_{yy}\dot{Q} + I_{xz}(P^2 - R^2) + (I_{xx} - I_{zz})PR \quad (2.31)$$

$$N = I_{zz}\dot{R} - I_{xz}(\dot{P} - QR) + (I_{yy} - I_{xx})PQ \quad (2.32)$$

### 2.2.4 Linearisasi

Karena persamaan gaya pada 2.12 - 2.14 dan persamaan momen torsi pada 2.30 - 2.32 merupakan persamaan nonlinear. Dalam hal ini komponen-komponen arah gaya, momen dan kecepatan dapat ditulis menjadi [10]:

$$\begin{aligned} X &= X_0 + \Delta X & U &= U_0 + \Delta U & P &= P_0 + \Delta P \\ Y &= Y_0 + \Delta Y & V &= V_0 + \Delta V & Q &= Q_0 + \Delta Q \\ Z &= Z_0 + \Delta Z & W &= W_0 + \Delta W & R &= R_0 + \Delta R \end{aligned}$$

$$\begin{aligned} L &= L_0 + \Delta L & \Theta &= \Theta_0 + \Delta \Theta \\ M &= M_0 + \Delta M & \Phi &= \Phi_0 + \Delta \Phi \\ N &= N_0 + \Delta N & \Psi &= \Psi_0 + \Delta \Psi \end{aligned}$$

Komponen-komponen tersebut terdiri dari keadaan setimbang (*trim*) dan gangguan.

Sehingga persamaan gaya pada 2.12 - 2.14 menjadi

$$\begin{aligned} X_0 + \Delta X &= m \left\{ \left( \dot{U}_0 + \Delta \dot{U} \right) + (Q_0 + \Delta Q) (W_0 + \Delta W) \right. \\ &\quad \left. - (R_0 + \Delta R) (V_0 + \Delta V) \right. \\ &\quad \left. + g \sin (\Theta_0 + \Delta \Theta) \right\} \end{aligned} \quad (2.33)$$

$$\begin{aligned} Y_0 + \Delta Y &= m \left\{ \left( \dot{V}_0 + \Delta \dot{V} \right) + (R_0 + \Delta R) (U_0 + \Delta U) \right. \\ &\quad \left. - (P_0 + \Delta P) (W_0 + \Delta W) \right. \\ &\quad \left. - g \cos (\Theta_0 + \Delta \Theta) \sin (\Phi_0 + \Delta \Phi) \right\} \end{aligned} \quad (2.34)$$

$$\begin{aligned} Z_0 + \Delta Z &= m \left\{ \left( \dot{W}_0 + \Delta \dot{W} \right) + (P_0 + \Delta P) (V_0 + \Delta V) \right. \\ &\quad \left. - (Q_0 + \Delta Q) (U_0 + \Delta U) \right. \\ &\quad \left. - g \cos (\Theta_0 + \Delta \Theta) \cos (\Phi_0 + \Delta \Phi) \right\} \end{aligned} \quad (2.35)$$

dan persamaan 2.33 - 2.35 dapat ditulis menjadi

$$\begin{aligned}
 X_0 + \Delta X &= m \left\{ \left( \dot{U}_0 + \Delta \dot{U} \right) + (Q_0 W_0 + Q_0 \Delta W + W_0 \Delta Q + \Delta Q \Delta W) \right. \\
 &\quad - (R_0 V_0 + R_0 \Delta V + V_0 \Delta R + \Delta R \Delta V) \\
 &\quad \left. + g (\sin \Theta_0 \cos \Delta \Theta + \cos \Theta_0 \sin \Delta \Theta) \right\} \quad (2.36)
 \end{aligned}$$

$$\begin{aligned}
 Y_0 + \Delta Y &= m \left\{ \left( \dot{V}_0 + \Delta \dot{V} \right) + (R_0 U_0 + R_0 \Delta U + U_0 \Delta R + \Delta R \Delta U) \right. \\
 &\quad - (P_0 W_0 + P_0 \Delta W + W_0 \Delta P + \Delta P \Delta W) \\
 &\quad - g (\cos \Theta_0 \cos \Delta \Theta - \sin \Theta_0 \sin \Delta \Theta) \\
 &\quad \left. (\sin \Phi_0 \cos \Delta \Phi + \cos \Phi_0 \sin \Delta \Phi) \right\} \quad (2.37)
 \end{aligned}$$

$$\begin{aligned}
 Z_0 + \Delta Z &= m \left\{ \left( \dot{W}_0 + \Delta \dot{W} \right) + (P_0 V_0 + P_0 \Delta V + V_0 \Delta P + \Delta P \Delta V) \right. \\
 &\quad - (Q_0 U_0 + Q_0 \Delta U + U_0 \Delta Q + \Delta Q \Delta U) \\
 &\quad - g (\cos \Theta_0 \cos \Delta \Theta - \sin \Theta_0 \sin \Delta \Theta) \\
 &\quad \left. (\cos \Phi_0 \cos \Delta \Phi - \sin \Phi_0 \sin \Delta \Phi) \right\} \quad (2.38)
 \end{aligned}$$

Sedangkan untuk persamaan momen torsi pada 2.30 - 2.32 menjadi

$$\begin{aligned}
 L_0 + \Delta L &= I_{xx} \left( \dot{P}_0 + \Delta \dot{P} \right) - I_{xz} \left( \left( \dot{R}_0 + \Delta \dot{R} \right) + \right. \\
 &\quad \left. (P_0 + \Delta P) (Q_0 + \Delta Q) \right) + \\
 &\quad (I_{zz} - I_{yy}) (Q_0 + \Delta Q) (U_0 + \Delta U) \quad (2.39)
 \end{aligned}$$

$$\begin{aligned}
M_0 + \Delta M &= I_{yy} \left( \dot{Q}_0 + \Delta \dot{Q} \right) + I_{xz} \left( (P_0 + \Delta P)^2 - \right. \\
&\quad \left. (R_0 + \Delta R)^2 \right) + (I_{xx} - I_{zz}) \\
&\quad (P_0 + \Delta P) (Q_0 + \Delta Q) \quad (2.40)
\end{aligned}$$

$$\begin{aligned}
N_0 + \Delta N &= I_{xx} \left( \dot{R}_0 + \Delta \dot{R} \right) - I_{xz} \left( \left( \dot{P}_0 + \Delta \dot{P} \right) + \right. \\
&\quad \left. (Q_0 + \Delta Q) (R_0 + \Delta R) \right) + \\
&\quad (I_{yy} - I_{xx}) (P_0 + \Delta P) (Q_0 + \Delta Q) \quad (2.41)
\end{aligned}$$

dan persamaan 2.39 - 2.41 dapat ditulis menjadi

$$\begin{aligned}
L_0 + \Delta L &= I_{xx} \left( \dot{P}_0 + \Delta \dot{P} \right) - I_{xz} \left( \left( \dot{R}_0 + \Delta \dot{R} \right) + \right. \\
&\quad \left. (P_0 Q_0 + P_0 \Delta Q + Q_0 \Delta P + \Delta P \Delta Q) \right) + \\
&\quad (I_{zz} - I_{yy}) \left( Q_0 R_0 + Q_0 \Delta R + \right. \\
&\quad \left. R_0 \Delta Q + \Delta Q \Delta R \right) \quad (2.42)
\end{aligned}$$

$$\begin{aligned}
M_0 + \Delta M &= I_{yy} \left( \dot{Q}_0 + \Delta \dot{Q} \right) + I_{xz} \left( P_0^2 + 2P_0 \Delta P + \right. \\
&\quad \left. (\Delta P)^2 - R_0^2 - 2R_0 \Delta R - (\Delta R)^2 \right) \\
&\quad + (I_{xx} - I_{zz}) \left( P_0 R_0 + P_0 \Delta R + \right. \\
&\quad \left. R_0 \Delta P + \Delta P \Delta R \right) \quad (2.43)
\end{aligned}$$

$$\begin{aligned}
N_0 + \Delta N &= I_{zz} \left( \dot{R}_0 + \Delta \dot{R} \right) - I_{xz} \left( \left( \dot{P}_0 + \Delta \dot{P} \right) + \right. \\
&\quad \left. (Q_0 R_0 + Q_0 \Delta R + R_0 \Delta Q + \Delta Q \Delta R) \right) + \\
&\quad (I_{yy} - I_{xx}) \left( P_0 Q_0 + P_0 \Delta Q + \right. \\
&\quad \left. Q_0 \Delta P + \Delta P \Delta Q \right) \tag{2.44}
\end{aligned}$$

Persamaan-persamaan pada 2.36 - 2.38 dan 2.42 - 2.44 terdiri dari kondisi setimbang (*trim*) dan adanya gangguan. Dengan mengasumsikan kondisi rata-rata gangguan sangat kecil, sehingga [10]:

- Sinus dari sudut gangguan dianggap sama dengan sudut gangguan
- Cosinus dari sudut gangguan dianggap sama dengan satu

Maka persamaaan 2.36 - 2.38 menjadi

$$\begin{aligned}
X_0 + \Delta X &= m \left\{ \dot{U}_0 + \Delta \dot{U} + (Q_0 W_0 + Q_0 \Delta W + W_0 \Delta Q + \Delta Q \Delta W) \right. \\
&\quad \left. - (R_0 V_0 + R_0 \Delta V + V_0 \Delta R + \Delta R \Delta V) \right. \\
&\quad \left. + g (\sin \Theta_0 + \Delta \Theta \cos \Theta_0) \right\} \tag{2.45}
\end{aligned}$$

$$\begin{aligned}
Y_0 + \Delta Y &= m \left\{ \dot{V}_0 + \Delta \dot{V} + (R_0 U_0 + R_0 \Delta U + U_0 \Delta R + \Delta R \Delta U) \right. \\
&\quad \left. - (P_0 W_0 + P_0 \Delta W + W_0 \Delta P + \Delta P \Delta W) \right. \\
&\quad \left. - g (\cos \Theta_0 - \Delta \Theta \sin \Theta_0) \right. \\
&\quad \left. (\sin \Phi_0 + \Delta \Phi \cos \Phi_0) \right\} \tag{2.46}
\end{aligned}$$

$$\begin{aligned}
Z_0 + \Delta Z = m \Big\{ & \dot{W}_0 + \Delta \dot{W} + (P_0 V_0 + P_0 \Delta V + V_0 \Delta P + \Delta P \Delta V) \\
& - (Q_0 U_0 + Q_0 \Delta U + U_0 \Delta Q + \Delta Q \Delta U) \\
& - g (\cos \Theta_0 - \Delta \Theta \sin \Theta_0) \\
& (\cos \Phi_0 - \Delta \Phi \sin \Phi_0) \Big\} \quad (2.47)
\end{aligned}$$

Pada persamaan 2.42 - 2.44 dan 2.45 - 2.47 merupakan persamaan gerak pesawat terbang pada saat kondisi setimbang (*trim*) dan persamaan gangguan. Karena linearisasi kondisi setimbang (*trim*) diabaikan. Maka persamaan 2.45 - 2.47 menjadi

$$\begin{aligned}
\Delta X = m \Big\{ & \Delta \dot{U} + Q_0 \Delta W + W_0 \Delta Q + \Delta Q \Delta W \\
& - R_0 \Delta V - V_0 \Delta R - \Delta R \Delta V + g \Delta \Theta \cos \Theta_0 \Big\} \quad (2.48)
\end{aligned}$$

$$\begin{aligned}
\Delta Y = m \Big\{ & \Delta \dot{V} + R_0 \Delta U + U_0 \Delta R + \Delta R \Delta U \\
& - P_0 \Delta W - W_0 \Delta P - \Delta P \Delta W - g \Big( \Delta \Phi \cos \Theta_0 \cos \Phi_0 \\
& - \Delta \Theta \sin \Theta_0 \sin \Phi_0 - \Delta \Theta \Delta \Phi \sin \Theta_0 \cos \Phi_0 \Big) \Big\} \quad (2.49)
\end{aligned}$$

$$\begin{aligned}
\Delta Z = m \Big\{ & \Delta \dot{W} + P_0 \Delta V + V_0 \Delta P + \Delta P \Delta V \\
& - Q_0 \Delta U - U_0 \Delta Q - \Delta Q \Delta U + g \Big( \Delta \Phi \cos \Theta_0 \sin \Phi_0 \\
& + \Delta \Theta \sin \Theta_0 \cos \Phi_0 - \Delta \Theta \Delta \Phi \sin \Theta_0 \sin \Phi_0 \Big) \Big\} \quad (2.50)
\end{aligned}$$



Maka persamaan 2.42 - 2.44 menjadi

$$\begin{aligned}\Delta L = & I_{xx}\Delta\dot{P} - I_{xz}\left(\Delta\dot{R} + P_0\Delta Q + \right. \\ & \left. Q_0\Delta P + \Delta P\Delta Q\right) + (I_{zz} - I_{yy}) \\ & (Q_0\Delta R + R_0\Delta Q + \Delta Q\Delta R)\end{aligned}\quad (2.51)$$

$$\begin{aligned}\Delta M = & I_{yy}\Delta\dot{Q} + I_{xz}\left(2P_0\Delta P + (\Delta P)^2 - \right. \\ & \left. - 2R_0\Delta R - (\Delta R)^2\right) + (I_{xx} - I_{zz}) \\ & (P_0\Delta R + R_0\Delta P + \Delta P\Delta R)\end{aligned}\quad (2.52)$$

$$\begin{aligned}\Delta N = & I_{zz}\Delta\dot{R} - I_{xz}\left(\Delta\dot{P} - Q_0\Delta R - \right. \\ & \left. R_0\Delta Q + \Delta Q\Delta R\right) + (I_{yy} - I_{xx}) \\ & (P_0\Delta Q + Q_0\Delta P + \Delta P\Delta Q)\end{aligned}\quad (2.53)$$

Pesawat diasumsikan terbang simetris, sehingga [10]

- Terbang simetri :  $V_0 = 0$
- Terbang dengan sayap mendatar :  $\Phi_0 = 0$
- Terbang setimbang :  $P_0 = Q_0 = R_0 = 0$

Sehingga persamaan 2.48 - 2.50 menjadi

$$\begin{aligned}\Delta X = & m\left(\Delta\dot{U} + W_0\Delta Q + \Delta Q\Delta W \right. \\ & \left. - \Delta R\Delta V + g\Delta\Theta \cos\Theta_0\right)\end{aligned}\quad (2.54)$$

$$\begin{aligned}\Delta Y = & m \left( \Delta \dot{V} + U_0 \Delta R + \Delta R \Delta U \right. \\ & \left. - W_0 \Delta P - \Delta P \Delta W - g \right. \\ & \left. (\Delta \Phi \cos \Theta_0 - \Delta \Theta \Delta \Phi \sin \Theta_0) \right) \quad (2.55)\end{aligned}$$

$$\begin{aligned}\Delta Z = & m \left( \Delta \dot{W} + \Delta P \Delta V - U_0 \Delta Q \right. \\ & \left. - \Delta Q \Delta U + g \Delta \Theta \sin \Theta_0 \right) \quad (2.56)\end{aligned}$$

dan persamaan 2.51 - 2.53 menjadi

$$\begin{aligned}\Delta L = & I_{xx} \Delta \dot{P} - I_{xz} \left( \Delta \dot{R} + \Delta P \Delta Q \right) \\ & + (I_{zz} - I_{yy}) \Delta Q \Delta R \quad (2.57)\end{aligned}$$

$$\begin{aligned}\Delta M = & I_{yy} \Delta \dot{Q} + I_{xz} \left( (\Delta P)^2 - (\Delta R)^2 \right) \\ & + (I_{xx} - I_{zz}) \Delta P \Delta R \quad (2.58)\end{aligned}$$

$$\begin{aligned}\Delta N = & I_{zz} \Delta \dot{R} - I_{xz} \left( \Delta \dot{P} - \Delta Q \Delta R \right) \\ & + (I_{yy} - I_{xx}) \Delta P \Delta Q \quad (2.59)\end{aligned}$$

Untuk menyederhanakan bentuk penulisan pada persamaan 2.54 - 2.56 dan 2.57 - 2.59 didefinisikan sebagai berikut :

$$\begin{aligned}\Delta X = x & \quad \Delta U = u & \Delta P = p & \quad \Delta L = l & \quad \Delta \Theta = \theta \\ \Delta Y = y & \quad \Delta V = v & \Delta Q = q & \quad \Delta M = m & \quad \Delta \Phi = \phi \\ \Delta Z = z & \quad \Delta W = w & \Delta R = r & \quad \Delta N = n & \quad \Delta \Psi = \psi\end{aligned}$$

Sehingga persamaan 2.54 - 2.56 menjadi

$$x = m (\dot{u} + W_0 q + q w - r v + g \theta \cos \Theta_0) \quad (2.60)$$

$$\begin{aligned}y = & m \left( \dot{v} + U_0 r + r u - W_0 p - p w \right. \\ & \left. - g (\phi \cos \Theta_0 - \theta \phi \sin \Theta_0) \right) \quad (2.61)\end{aligned}$$

$$z = m (\dot{w} + p v - U_0 q - q u + g \theta \sin \Theta_0) \quad (2.62)$$

dan persamaan 2.57 - 2.59

$$l = I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) + (I_{zz} - I_{yy})qr \quad (2.63)$$

$$m = I_{yy}\dot{q} + I_{xz}(p^2 - r^2) + (I_{xx} - I_{zz})pr \quad (2.64)$$

$$n = I_{zz}\dot{r} - I_{xz}(\dot{p} - qr) + (I_{yy} - I_{xx})pq \quad (2.65)$$

Persamaan-persamaan tersebut merupakan persamaan perubahan kecil di sekitar nilai kesetimbangan. Adapun fungsi gangguan yang cukup dominan dalam analisis gerak pesawat terbang adalah sebagai berikut [10]:

$$x = f(u, w, \delta_e, \delta_T)$$

$$y = f(v, p, r, \delta_r)$$

$$z = f(u, w, \dot{w}, q, \delta_e, \delta_T)$$

$$l = f(v, p, r, \delta_r, \delta_a)$$

$$m = f(u, w, \dot{w}, q, \delta_e, \delta_T)$$

$$n = f(v, p, r, \delta_r, \delta_a)$$

Dengan menggunakan deret Taylor persamaan fungsi gangguan gaya, sebagai berikut [10]:

$$x = \frac{\partial X}{\partial u}u + \frac{\partial X}{\partial w}w + \frac{\partial X}{\partial \delta_e}\delta_e + \frac{\partial X}{\partial \delta_T}\delta_T \quad (2.66)$$

$$y = \frac{\partial Y}{\partial v}v + \frac{\partial Y}{\partial p}p + \frac{\partial Y}{\partial r}r + \frac{\partial Y}{\partial \delta_r}\delta_r \quad (2.67)$$

$$\begin{aligned} z = & \frac{\partial Z}{\partial u}u + \frac{\partial Z}{\partial w}w + \frac{\partial Z}{\partial \dot{w}}\dot{w} + \frac{\partial Z}{\partial q}q + \frac{\partial Z}{\partial \delta_e}\delta_e \\ & + \frac{\partial Z}{\partial \delta_T}\delta_T \end{aligned} \quad (2.68)$$

dan untuk persamaan fungsi gangguan momen torsi, sebagai berikut [10]:

$$l = \frac{\partial L}{\partial v}v + \frac{\partial L}{\partial p}p + \frac{\partial L}{\partial r}r + \frac{\partial L}{\partial \delta_r}\delta_r + \frac{\partial L}{\partial \delta_a}\delta_a \quad (2.69)$$

$$\begin{aligned} m = & \frac{\partial M}{\partial u}u + \frac{\partial M}{\partial w}w + \frac{\partial M}{\partial \dot{w}}\dot{w} + \frac{\partial M}{\partial q}q + \frac{\partial M}{\partial \delta_e}\delta_e \\ & + \frac{\partial M}{\partial \delta_T}\delta_T \end{aligned} \quad (2.70)$$

$$n = \frac{\partial N}{\partial v}v + \frac{\partial N}{\partial p}p + \frac{\partial N}{\partial r}r + \frac{\partial N}{\partial \delta_r}\delta_r + \frac{\partial N}{\partial \delta_a}\delta_a \quad (2.71)$$

Substitusi persamaan 2.60 - 2.62 ke persamaan 2.66 - 2.68 sehingga diperoleh

$$\begin{aligned} m(\dot{u} + W_0q + qw - rv + g\theta \cos \Theta_0) = & \frac{\partial X}{\partial u}u + \frac{\partial X}{\partial w}w + \frac{\partial X}{\partial \delta_e}\delta_e \\ & + \frac{\partial X}{\partial \delta_T}\delta_T \end{aligned} \quad (2.72)$$

$$\begin{aligned} m(\dot{v} + U_0r + ru - W_0p - pw - g(\phi \cos \Theta_0 - \theta \phi \sin \Theta_0)) = \\ \frac{\partial Y}{\partial v}v + \frac{\partial Y}{\partial p}p + \frac{\partial Y}{\partial r}r + \frac{\partial Y}{\partial \delta_r}\delta_r \end{aligned} \quad (2.73)$$

$$\begin{aligned} m(\dot{w} + pv - U_0q - qu + g\theta \sin \Theta_0) = & \frac{\partial Z}{\partial u}u + \frac{\partial Z}{\partial w}w + \frac{\partial Z}{\partial \dot{w}}\dot{w} \\ & + \frac{\partial Z}{\partial q}q + \frac{\partial Z}{\partial \delta_e}\delta_e + \frac{\partial Z}{\partial \delta_T}\delta_T \end{aligned} \quad (2.74)$$

Substitusi persamaan 2.63 - 2.65 ke persamaan 2.69 - 2.71

$$\begin{aligned} I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) + (I_{zz} - I_{yy})qr = & \frac{\partial L}{\partial v}v \\ & + \frac{\partial L}{\partial p}p + \frac{\partial L}{\partial r}r + \frac{\partial L}{\partial \delta_r}\delta_r + \frac{\partial L}{\partial \delta_a}\delta_a \end{aligned} \quad (2.75)$$

$$I_{yy}\dot{q} + I_{xz}(p^2 - r^2) + (I_{xx} - I_{zz})pr = \frac{\partial M}{\partial u}u + \frac{\partial M}{\partial w}w + \frac{\partial M}{\partial \dot{w}}\dot{w} + \frac{\partial M}{\partial q}q + \frac{\partial M}{\partial \delta_e}\delta_e + \frac{\partial M}{\partial \delta_T}\delta_T \quad (2.76)$$

$$I_{zz}\dot{r} - I_{xz}(\dot{p} - qr) + (I_{yy} - I_{xx})pq = \frac{\partial N}{\partial v}v + \frac{\partial N}{\partial p}p + \frac{\partial N}{\partial r}r + \frac{\partial N}{\partial \delta_r}\delta_r + \frac{\partial N}{\partial \delta_a}\delta_a \quad (2.77)$$

Persamaan 2.72 - 2.74 dapat ditulis menjadi

$$\begin{aligned} \dot{u} = & \frac{1}{m} \frac{\partial X}{\partial u}u + \frac{1}{m} \frac{\partial X}{\partial w}w + \frac{1}{m} \frac{\partial X}{\partial \delta_e}\delta_e + \frac{1}{m} \frac{\partial X}{\partial \delta_T}\delta_T \\ & - W_0q - qw + rv - g\theta \cos \Theta_0 \end{aligned} \quad (2.78)$$

$$\begin{aligned} \dot{v} = & \frac{1}{m} \frac{\partial Y}{\partial v}v + \frac{1}{m} \frac{\partial Y}{\partial p}p + \frac{1}{m} \frac{\partial Y}{\partial r}r + \frac{1}{m} \frac{\partial Y}{\partial \delta_r}\delta_r - U_0r - ru \\ & + W_0p + pw + g(\phi \cos \Theta_0 - \theta \phi \sin \Theta_0) \end{aligned} \quad (2.79)$$

$$\begin{aligned} \dot{w} = & \frac{1}{m} \frac{\partial Z}{\partial u}u + \frac{1}{m} \frac{\partial Z}{\partial w}w + \frac{1}{m} \frac{\partial Z}{\partial \dot{w}}\dot{w} + \frac{1}{m} \frac{\partial Z}{\partial q}q + \frac{1}{m} \frac{\partial Z}{\partial \delta_e}\delta_e \\ & + \frac{1}{m} \frac{\partial Z}{\partial \delta_T}\delta_T - pv + U_0q + qu - g\theta \sin \Theta_0 \end{aligned} \quad (2.80)$$

Persamaan 2.75 - 2.77 dapat ditulis menjadi

$$\begin{aligned} \dot{p} = & \frac{1}{I_{xx}} \frac{\partial L}{\partial v}v + \frac{1}{I_{xx}} \frac{\partial L}{\partial p}p + \frac{1}{I_{xx}} \frac{\partial L}{\partial r}r + \frac{1}{I_{xx}} \frac{\partial L}{\partial \delta_r}\delta_r + \frac{1}{I_{xx}} \frac{\partial L}{\partial \delta_a}\delta_a \\ & + \frac{I_{xz}}{I_{xx}}(\dot{r} + pq) - \frac{(I_{zz} - I_{yy})}{I_{xx}}qr \end{aligned} \quad (2.81)$$

$$\begin{aligned} \dot{q} = & \frac{1}{I_{yy}} \frac{\partial M}{\partial u}u + \frac{1}{I_{yy}} \frac{\partial M}{\partial w}w + \frac{1}{I_{yy}} \frac{\partial M}{\partial \dot{w}}\dot{w} + \frac{1}{I_{yy}} \frac{\partial M}{\partial q}q + \frac{1}{I_{yy}} \frac{\partial M}{\partial \delta_e}\delta_e \\ & + \frac{1}{I_{yy}} \frac{\partial M}{\partial \delta_T}\delta_T - \frac{I_{xz}}{I_{yy}}(p^2 - r^2) - \frac{(I_{xx} - I_{zz})}{I_{yy}}pr \end{aligned} \quad (2.82)$$

$$\begin{aligned} \dot{r} = & \frac{1}{I_{zz}} \frac{\partial N}{\partial v}v + \frac{1}{I_{zz}} \frac{\partial N}{\partial p}p + \frac{1}{I_{zz}} \frac{\partial N}{\partial r}r + \frac{1}{I_{zz}} \frac{\partial N}{\partial \delta_r}\delta_r + \frac{1}{I_{zz}} \frac{\partial N}{\partial \delta_a}\delta_a \\ & + \frac{I_{xz}}{I_{zz}}(\dot{p} - qr) - \frac{(I_{yy} - I_{xx})}{I_{zz}}pq \end{aligned} \quad (2.83)$$

Dengan mendefinisikan :

$$\begin{aligned} X_i &= \frac{1}{m} \frac{\partial X}{\partial i} \\ Y_i &= \frac{1}{m} \frac{\partial Y}{\partial i} \\ Z_i &= \frac{1}{m} \frac{\partial Z}{\partial i} \end{aligned} \quad \text{dan} \quad \begin{aligned} L_i &= \frac{1}{I_{xx}} \frac{\partial L}{\partial i} \\ M_i &= \frac{1}{I_{yy}} \frac{\partial M}{\partial i} \\ N_i &= \frac{1}{I_{zz}} \frac{\partial N}{\partial i} \end{aligned}$$

Sehingga persamaan 2.78 - 2.80 dan 2.81 - 2.83 menjadi

$$\begin{aligned} \dot{u} &= X_u u + X_w w + X_{\delta_e} \delta_e + X_{\delta_T} \delta_T - W_0 q - q w \\ &\quad + r v - g \theta \cos \Theta_0 \\ \dot{v} &= Y_v v + Y_p p + Y_r r + Y_{\delta_r} \delta_r - U_0 r - r u + W_0 p + p w \\ &\quad + g (\phi \cos \Theta_0 - \theta \phi \sin \Theta_0) \\ \dot{w} &= Z_u u + Z_w w + Z_{\dot{w}} \dot{w} + Z_q q + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T \\ &\quad - p v + U_0 q + q u - g \theta \sin \Theta_0 \\ \dot{p} &= L_v v + L_p p + L_r r + L_{\delta_r} \delta_r + L_{\delta_a} \delta_a \\ &\quad + \frac{I_{xz}}{I_{xx}} (\dot{r} + p q) - \frac{(I_{zz} - I_{yy})}{I_{xx}} q r \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta_e} \delta_e + M_{\delta_T} \delta_T \\ &\quad - \frac{I_{xz}}{I_{yy}} (p^2 - r^2) - \frac{(I_{xx} - I_{zz})}{I_{yy}} p r \\ \dot{r} &= N_v v + N_p p + N_r r + N_{\delta_r} \delta_r + N_{\delta_a} \delta_a \\ &\quad + \frac{I_{xz}}{I_{zz}} (\dot{p} - q r) - \frac{(I_{yy} - I_{xx})}{I_{zz}} p q \end{aligned}$$

### 2.2.5 Persamaan Gerak Longitudinal Pesawat Terbang

Gerak longitudinal pesawat terbang adalah gerak sepanjang sumbu  $x$  dimana sudut *pitch* dikontrol oleh *defleksi elevator* dan *defleksi throttle*. Pada persamaan gerak longitudinal pesawat terbang, parameter  $Z_q q$ ,  $Z_{\dot{w}} \dot{w}$  diabaikan karena tidak berpengaruh signifikan terhadap respon pesawat terbang pada gerak longitudinal [10]. Dan untuk sumbu kestabilan pada saat kondisi setimbang sumbu  $x$  terhadap CG

*center of gravity* merupakan sumbu longitudinal dari pesawat terbang sehingga mengakibatkan  $W_0 = 0$ . Dan juga karena sudut *pitch* pada saat setimbang ( $\Theta_0$ ) relatif kecil, dimana  $\Theta_0 \approx 0$ , maka  $\cos \Theta_0 = 1$  dan  $\sin \Theta_0 = 0$ .

Pada gerak longitudinal merupakan gerak yang diperoleh dari gaya-gaya yang bekerja pada bidang  $xz$ . Gerak longitudinal dipengaruhi oleh kecepatan translasi ke depan, kecepatan translasi ke atas, kecepatan sudut *pitch*.

Sehingga persamaan gerak longitudinal pesawat terbang, sebagai berikut :

$$\begin{aligned}\dot{u} &= X_u u + X_w w + X_{\delta_e} \delta_e + X_{\delta_T} \delta_T - qw + rv - g\theta \\ \dot{w} &= Z_u u + Z_w w + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T - pv + U_0 q + qu \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta_e} \delta_e + M_{\delta_T} \delta_T \\ &\quad - \frac{I_{xz}}{I_{yy}} (p^2 - r^2) - \frac{(I_{xx} - I_{zz})}{I_{yy}} pr\end{aligned}\quad (2.84)$$

Untuk parameter-parameter  $rv, p^2, r^2, pr$  merupakan persamaan pada gerak transversal pesawat terbang sehingga parameter-parameter tersebut diabaikan.

Sehingga persamaan 2.84 menjadi

$$\begin{aligned}\dot{u} &= X_u u + X_w w - wq - g\theta + X_{\delta_e} \delta_e + X_{\delta_T} \delta_T \\ \dot{w} &= Z_u u + Z_w w + U_0 q + qu + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta_e} \delta_e + M_{\delta_T} \delta_T\end{aligned}$$

Dan untuk memenuhi persamaan gerak longitudinal pesawat terbang, dengan menambahkan variabel sudut *pitch* ( $\theta$ ). Maka persamaan gerak longitudinal pesawat terbang dipengaruhi oleh kecepatan translasi ke depan, kecepatan translasi ke atas, kecepatan sudut *pitch* dan sudut *pitch*.

Sehingga persamaan gerak longitudinal pesawat terbang

menjadi sebagai berikut :

$$\begin{aligned}
 \dot{u} &= X_u u + X_w w - wq - g\theta + X_{\delta_e} \delta_e + X_{\delta_T} \delta_T \\
 \dot{w} &= Z_u u + Z_w w + U_0 q + qu + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T \\
 \dot{q} &= M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta_e} \delta_e + M_{\delta_T} \delta_T \\
 \dot{\theta} &= q
 \end{aligned} \tag{2.85}$$

Dan dapat ditulis menjadi

$$\begin{aligned}
 \dot{q} &= M_u u + M_w w + M_{\dot{w}} (Z_u u + Z_w w + U_0 q + qu \\
 &\quad + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T) + M_q q + M_{\delta_e} \delta_e + M_{\delta_T} \delta_T \\
 &= (M_u + M_{\dot{w}} Z_u) u + (M_w + M_{\dot{w}} Z_w) w + (M_q + M_{\dot{w}} U_0) q \\
 &\quad + M_{\dot{w}} qu + (M_{\delta_e} + M_{\dot{w}} Z_{\delta_e}) \delta_e + (M_{\delta_T} + M_{\dot{w}} Z_{\delta_T}) \delta_T
 \end{aligned}$$

Sehingga persamaan 2.85 menjadi

$$\begin{aligned}
 \dot{u} &= X_u u + X_w w - wq - g\theta + X_{\delta_e} \delta_e + X_{\delta_T} \delta_T \\
 \dot{w} &= Z_u u + Z_w w + U_0 q + qu + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T \\
 \dot{q} &= (M_u + M_{\dot{w}} Z_u) u + (M_w + M_{\dot{w}} Z_w) w + (M_q + M_{\dot{w}} U_0) q \\
 &\quad + M_{\dot{w}} qu + (M_{\delta_e} + M_{\dot{w}} Z_{\delta_e}) \delta_e + (M_{\delta_T} + M_{\dot{w}} Z_{\delta_T}) \delta_T \\
 \dot{\theta} &= q
 \end{aligned} \tag{2.86}$$

## 2.3 Fuzzy Kalman Filter

### 2.3.1 Sistem Fuzzy

Lotfi A. Zadeh (1965) merupakan orang pertama kali yang memperkenalkan teori himpunan Fuzzy sebagai bentuk permasalahan dalam hal ketidakpastian. Teori Fuzzy dapat digunakan untuk mengkontruksi hubungan nonlinear dengan informasi heuristik. Dalam konteks Fuzzy, himpunan *crisp* didefinisikan sebagai himpunan yang memiliki elemen-elemen yang pasti dan dapat dibedakan. Lotfi A. Zadeh menulis " Konsep dari himpunan Fuzzy memberikan titik mudah keberangkatan untuk mengkonstruksi suatu kerangka kerja



konseptual yang sejajar dengan berbagai aspek kerangka yang digunakan dalam hal himpunan biasa, namun lebih umum daripada sebelumnya serta berpotensi, bisa terbukti memiliki ruang lengkap yang lebih luas dari penerapan, khususnya di bidang klasifikasi pola dan pengolahan informasi. Pada dasarnya, kerangka tersebut menyediakan cara alami menangani masalah yang menjadi sumber ketidaktepatan adalah belum adanya kriteria yang didefinisikan secara tajam dari kelas keanggotaan bukan pada adanya variabel acak”.

Sistem inferensi Fuzzy merupakan suatu bentuk kerangka yang menganut aturan pada teori himpunan Fuzzy. Dimana aturan dasar sistem inferensi Fuzzy yaitu berbentuk IF-THEN. Dengan demikian, jika kondisi ”Diberikan”, maka kesimpulannya adalah ”Tersirat”. Salah satu metode sistem inferensi Fuzzy yaitu metode Sugeno. Metode ini diperkenalkan pertama kali oleh Michio Sugeno. Adapun tahapan-tahapan metode Sugeno sebagai berikut [14]:

- Fuzzifikasi  
Proses fuzzifikasi merupakan tahap membentuk himpunan Fuzzy pada variabel-variabel Fuzzy dan menentukan fungsi keanggotaan.
- Aturan Dasar  
Aturan dasar logika Fuzzy untuk sistem inferensi Fuzzy Sugeno adalah

$$\begin{aligned} & \text{if}((x_1 \text{ is } A_1) \text{ AND } (x_2 \text{ is } A_2)) \dots \text{AND} (x_n \text{ is } A_n)) \\ & \text{THEN} (c = p_1 * x_1 + p_2 * x_2 + \dots + p_n * x_n / q) \end{aligned}$$

- Defuzzifikasi  
Proses defuzzifikasi merupakan tahap untuk bobot rata-

rata yang dihitung dengan cara :

$$C^* = \frac{\sum_{r=1}^R \alpha_r c_r}{\sum_{r=1}^R \alpha_r}$$

dimana  $c_r$  adalah output pada aturan dasar ke- $r$ ,  $\alpha_r$  merupakan himpunan-himpunan dari kombinasi fungsi keanggotaan dari variabel-variabel Fuzzy.

### 2.3.2 Fungsi Keanggotaan

Untuk himpunan fuzzy, fungsi karakteristik memungkinkan berbagai tingkatan keanggotaan untuk elemen dari himpunan.

*Definisi 2.1* [7]:

Jika  $X$  adalah kumpulan dari objek-objek secara umum dilambangkan oleh  $x$  maka himpunan Fuzzy  $\tilde{A}$  dalam  $X$  adalah himpunan pasangan yang berurutan :

$$\tilde{A} = \{ (x, \mu_{\tilde{A}}(x)) \mid x \in X \}$$

$\mu_{\tilde{A}}(x)$  disebut sebagai fungsi keanggotaan (*membership function*) atau kelas keanggotaan (tingkat komabilitas atau tingkat kebenaran) dari  $x$  di  $\tilde{A}$  yang dipetakan ke ruang keanggotaan. Hasil fungsi keanggotaan merupakan himpunan bagian dari bilangan real tak negatif yang supremum terbatas.

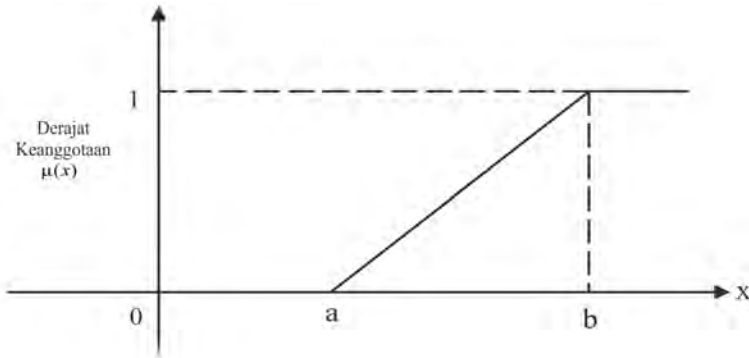
Fungsi keanggotaan (*membership function*) adalah suatu fungsi yang menunjukkan titik-titik input data ke dalam derajat keanggotaan. Representasi fungsi keanggotaan yang digunakan oleh Hasnawati Mahmuri (2011) yaitu representasi linear. Pada representasi linear, pemetaan input ke dalam derajat keanggotaannya digambarkan sebagai suatu garis lurus. Terdapat dua kondisi himpunan Fuzzy pada representasi linear [4],[9].

- Linear Naik

Representasi linear naik menggambarkan bahwa nilai

domain yang memiliki derajat keanggotaan nol bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

Fungsi Keanggotaan :



Gambar 2.4: Linear Naik

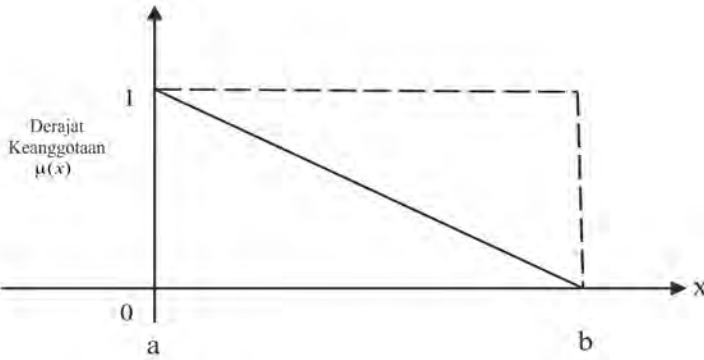
$$\mu_x = \begin{cases} 0 & ; x < a \\ \frac{x-a}{b-a} & ; a \leq x \leq b \\ 1 & ; x > b \end{cases}$$

- Linear Turun

Representasi linear turun menggambarkan bahwa nilai domain yang memiliki derajat keanggotaan tertinggi pada sisi kiri dan bergerak menurun ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

Fungsi Keanggotaan :

$$\mu_x = \begin{cases} \frac{b-x}{b-a} & ; a \leq x \leq b \\ 0 & ; x > b \end{cases}$$



Gambar 2.5: Linear Turun

### 2.3.3 Fuzzifikasi

Fuzzifikasi adalah proses pembuatan kuantitas Fuzzy. Jika bentuk ketidakpastian yang terjadi muncul karena ketidaktepatan, maka variabel Fuzzy dapat diwakili oleh suatu fungsi keanggotaan.

Diberikan sistem model dinamik, sebagai berikut [8]:

$$\begin{aligned}x_{k+1} &= A_k x_k + B_x u_x + G_k w_k \\ z_k &= H_k x_k + v_k\end{aligned}$$

Dengan  $x_k \in R^n$  dan  $z_k \in R^n$ , masing-masing model sistem keadaan dan model pengukuran. Fuzzy yang sesuai dengan matriks  $A_k$ ,  $B_k$ ,  $G_k$  dan  $H_k$  tersebut kemudian diberikan oleh :

$$\begin{aligned}A_f^i &= \{a_{pq} \in A \mid a_{pq} \text{ mempunyai fungsi keanggotaan } \mu_A^i(\cdot)\} \\ B_f^i &= \{b_{pq} \in B \mid b_{pq} \text{ mempunyai fungsi keanggotaan } \mu_B^i(\cdot)\} \\ G_f^i &= \{g_{pq} \in G \mid g_{pq} \text{ mempunyai fungsi keanggotaan } \mu_G^i(\cdot)\} \\ H_f^i &= \{h_{pq} \in H \mid h_{pq} \text{ mempunyai fungsi keanggotaan } \mu_H^i(\cdot)\}\end{aligned}$$

Dimana  $\mu_A^i(\cdot)$ ,  $\mu_B^i(\cdot)$ ,  $\mu_G^i(\cdot)$  dan  $\mu_H^i(\cdot)$  merupakan dalam fase aturan dasar logika Fuzzy [11].

### 2.3.4 Aturan Dasar Logika Fuzzy

Untuk menggambarkan aturan dasar logika Fuzzy IF-THEN,  $x_{k+1} = Ax_k$ , dimana dalam kasus ini  $A$  merupakan interval skalar. Pada kasus non Fuzzy, ketika datum,  $a$ , akan masuk ke dalam interval  $A$ , sehingga diperoleh  $x_{k+1} = ax_k$ . Jika terdapat fungsi keanggotaan sebanyak  $n$  yang terdefinisi pada interval  $A$ , maka datum  $a$  mempunyai nilai keanggotaan sebanyak  $n$  pula yaitu :  $\mu_A^1(a), \mu_A^2(a), \dots, \mu_A^n(a)$ . Semua nilai keanggotaan tersebut diperoleh sesuai kondisi parameter yang ada ketika (minimum dan maksimum). Sehingga misalkan mempunyai fungsi keanggotaan  $\mu_A^i(a)$  yang sesuai dengan hasil perhitungan  $x_{k+1}^i$  maka bergantung dari keadaan sebelumnya dapat dinyatakan sebagai berikut [11]:

$$x_{k+1}^i = \mu_A^i(a)ax_k$$

dengan  $x_k$  merupakan langkah sebelumnya. Secara umum, aturan dasar logika Fuzzy IF-THEN diberikan secara simbolis sebagai berikut [10]:

$$\text{Rule : IF } a \text{ is } A_i \text{ THEN } x_{k+1}^i = \mu_A^i(a)ax_k$$

Dengan  $a$  adalah  $A_i$  singkatan dari  $a$  yang dimiliki oleh  $A$  dan memiliki nilai keanggotaan  $\mu_A^i(a)$  dimana  $i = 1, 2, \dots, n$ .

### 2.3.5 Defuzzifikasi

Defuzzifikasi merupakan suatu proses filter untuk memperoleh hasil akhir estimasi yang terbaik. Defuzzifikasi dapat diartikan sebagai proses perubahan yang disajikan dalam bentuk himpunan-himpunan Fuzzy berdasarkan sesuai fungsi keanggotaannya untuk mendapatkan suatu nilai. Proses defuzzifikasi terbentuk berdasarkan aturan dasar (*rule*) yang mempunyai fungsi keanggotaan masing-masing, sehingga menjadi himpunan-himpunan Fuzzy yang akan digunakan sebagai acuan untuk memperoleh nilai hasil

estimasi. Sesuai prosedur standar defuzzifikasi, maka hasil akhir dari fase output dihitung menggunakan rumus bobot rata-rata, sebagai berikut [11]:

$$x_{k+1} = \frac{\rho^1 x_{k+1}^1 + \rho^2 x_{k+1}^2 + \dots + \rho^n x_{k+1}^n}{\rho^1 + \rho^2 + \dots + \rho^n} \quad (2.87)$$

dimana bobot  $\rho^i$  ( $i = 1, 2, \dots, n$ ) ditentukan sesuai aturan, dengan nilai keanggotaan dari input yang sesuai yaitu  $\rho^i = \mu_A^i(a)$  dimana  $i = 1, 2, \dots, n$  untuk menilai dengan kriteria optimal. Proses defuzzifikasi akan menghasilkan suatu hasil estimasi yang tunggal pada setiap waktu.

### 2.3.6 Kalman Filter

Kalman filter merupakan suatu estimator sistem dinamik linear. Kalman filter mampu mengestimasi variabel keadaan dinamik dari sistem dengan dua tahapan yaitu tahap prediksi dan tahap koreksi. Tahap prediksi (*time update*) merupakan tahap estimasi dari sistem model dinamik, sedangkan tahap koreksi (*measurement update*) merupakan tahap estimasi dari model pengukuran. Algoritma Kalman Filter waktu diskrit ditulis sebagai berikut [8]:

Model sistem dan model pengukuran :

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k \quad (2.88)$$

$$z_k = H_k x_k + v_k \quad (2.89)$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), w_k \sim N(0, Q_k), v_k \sim N(0, R_k)$$

Asumsi :

$\{w_k\}$  dan  $\{v_k\}$  adalah proses *white noise* yang tidak berkorelasi dengan  $x_0$  dan yang lainnya.  $R_k > 0$ .

Inisialisasi :

$$P_0 = P_{x_0}, \hat{x}_0 = \bar{x}_0$$

Tahap prediksi (*Time Update*) :

$$\begin{aligned} \text{kovarian error} &: P_{\bar{k}+1} = A_k P_k A_k^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{\bar{k}+1} = A_k \hat{x}_k + B_k u_k \end{aligned}$$

Tahap koreksi (*Measurement Update*) :

$$\begin{aligned} \text{kalmán gain} &: K_{k+1} = P_{\bar{k}+1} H_{k+1}^T (H_{k+1} P_{\bar{k}+1} H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1} = (I - K_{k+1} H_{k+1}) P_{\bar{k}+1} \\ \text{estimasi} &: \hat{x}_{k+1} = \hat{x}_{\bar{k}+1} + K_{k+1} (z_{k+1} - H_{k+1} \hat{x}_{\bar{k}+1}) \end{aligned}$$

### 2.3.7 Algoritma Fuzzy Kalman Filter

Metode kombinasi Logika Fuzzy dan Kalman Filter merupakan suatu metode yang telah diimplementasikan di berbagai permasalahan. Berdasarkan penelitian sebelumnya, kombinasi Logika Fuzzy dan Kalman Filter telah memberikan hasil estimasi yang lebih akurat daripada hanya menggunakan estimator Kalman Filter. Kombinasi ini disebut Fuzzy Kalman Filter.

Algoritma Fuzzy Kalman Filter penerapannya hampir sama dengan algoritma Kalman Filter. Namun, dalam algoritma Fuzzy Kalman Filter terdapat sebuah aturan (*rule*). Sesuai proses fuzzifikasi dengan aturan dasar logika Fuzzy dan dilakukan proses defuzzifikasi untuk memperoleh hasil akhir estimasi dengan fungsi bobot, sehingga algoritma Fuzzy Kalman Filter dapat ditulis sebagai berikut :

Model sistem dan model pengukuran [11]:

$$\begin{aligned} x_{k+1}^i &= A_k^i x_k + B_k^i u_k + G_k^i w_k \\ z_k^i &= H_k^i x_k + v_k \end{aligned}$$

Dengan  $i$  adalah *rule* ke- $i = 1, 2, \dots, n$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), w_k \sim N(0, Q_k), v_k \sim N(0, R_k)$$

Asumsi :

$\{w_k\}$  dan  $\{v_k\}$  adalah proses *white noise* yang tidak berkorelasi dengan  $x_0$  dan yang lainnya.  $R_k > 0$ .

Inisialisasi :

$$P_0 = P_{x_0}, \hat{x}_0 = \bar{x}_0$$

Tahap prediksi (*Time Update*) :

$$\begin{aligned} \text{kovarian error} &: P_{i,\bar{k}+1} = A_k^i P_{i,k} (A_k^i)^T + G_k^i Q_{i,k} (G_k^i)^T \\ \text{estimasi} &: \hat{x}_{i,\bar{k}+1} = A_k^i \hat{x}_{i,k} + B_k^i u_{i,k} \end{aligned}$$

Tahap koreksi (*Measurement Update*) :

$$\begin{aligned} \text{kalman gain} &: K_{i,k+1} = \frac{P_{i,\bar{k}+1} (H_{k+1}^i)^T}{(H_{k+1}^i P_{i,\bar{k}+1} (H_{k+1}^i)^T + R_{i,k+1})^{-1}} \\ \text{kovarian error} &: P_{i,k+1} = (I - K_{i,k+1} H_{k+1}^i) P_{i,\bar{k}+1} \\ \text{estimasi} &: \hat{x}_{i,k+1} = \hat{x}_{i,\bar{k}+1} + K_{i,k+1} (z_{k+1} - H_{k+1}^i \hat{x}_{i,\bar{k}+1}) \end{aligned}$$



## **BAB III**

### **METODE PENELITIAN**

Pada bab ini diuraikan tahapan-tahapan sistematis yang dilakukan dalam proses pengerjaan Tugas Akhir. Tahapan penelitian dalam Tugas Akhir ini terdiri atas empat tahap, yaitu studi literatur; mengkaji model gerak pesawat terbang, sistem Fuzzy dan Kalman Filter; implementasi metode; simulasi dan pembahasan; penarikan kesimpulan; pembuatan laporan tugas akhir.

#### **3.1 Studi Literatur**

Pada tahap ini dilakukan studi referensi tentang model gerak longitudinal pesawat terbang, fuzzifikasi, fungsi keanggotaan, defuzzifikasi, Algoritma Kalman Filter dan Fuzzy Kalman Filter. Referensi yang digunakan adalah buku-buku, skripsi, thesis dan paper-paper dalam jurnal ilmiah yang berkaitan dengan topik pada Tugas Akhir ini.

#### **3.2 Mengkaji model gerak pesawat terbang, sistem Fuzzy dan Kalman Filter**

Pada tahap ini akan dilakukan pengkajian mengenai model gerak longitudinal pesawat terbang. Model gerak pesawat terbang merupakan model nonlinear dan akan dibentuk model *state space* yang selanjutnya dilakukan pendiskritan. Selanjutnya akan dilakukan pengkajian mengenai sistem Fuzzy diantaranya fungsi keanggotaan yang digunakan untuk menentukan variabel-variabel pada saat minimum dan maksimum dalam gerak pesawat terbang, proses fuzzifikasi untuk menerapkan fungsi keanggotaan pada sistem model gerak pesawat terbang, aturan dasar logika Fuzzy dan proses

defuzzifikasi untuk mendapatkan hasil estimasi. Selanjutnya akan dilakukan pengkajian mengenai algoritma Kalman Filter dan penerapan Fuzzy Kalman Filter dengan menentukan defuzzifikasi untuk keluaran estimasi.

### 3.3 Implementasi Metode

Pada tahap ini dilakukan implementasi model gerak pesawat terbang dengan metode Kalman Filter dan Fuzzy Kalman Filter.

#### 3.3.1 Implementasi Kalman Filter

Metode Kalman Filter digunakan untuk sistem model linear. Adapun langkah-langkah yang dilakukan untuk estimasi gerak pesawat terbang dengan menggunakan metode Kalman Filter adalah sebagai berikut :

- (a) Menentukan Model Sistem dan Model Pengukuran  
Berdasarkan persamaan 2.88 - 2.89 diperoleh model sistem dan model pengukuran sebagai berikut:

$$\begin{aligned}x_{k+1} &= A_k x_k + B_k u_k + G_k w_k \\ z_k &= H_k x_k + v_k\end{aligned}$$

$$w_k \sim N(0, Q_k) \text{ dan } v_k \sim N(0, R_k)$$

dengan  $x = [u, w, q, \theta]^T$

- (b) Pelinearan

Model gerak longitudinal pesawat terbang merupakan model nonlinear sehingga dilakukan proses pelinearan dengan menggunakan Metode Jacobian yaitu :

$$J = \begin{bmatrix} \frac{\partial x_1}{\partial u} & \frac{\partial x_1}{\partial w} & \frac{\partial x_1}{\partial q} & \frac{\partial x_1}{\partial \theta} \\ \frac{\partial x_2}{\partial u} & \frac{\partial x_2}{\partial w} & \frac{\partial x_2}{\partial q} & \frac{\partial x_2}{\partial \theta} \\ \frac{\partial x_3}{\partial u} & \frac{\partial x_3}{\partial w} & \frac{\partial x_3}{\partial q} & \frac{\partial x_3}{\partial \theta} \\ \frac{\partial x_4}{\partial u} & \frac{\partial x_4}{\partial w} & \frac{\partial x_4}{\partial q} & \frac{\partial x_4}{\partial \theta} \end{bmatrix}$$

## (c) Pendiskritan

Metode Kalman Filter yang digunakan yaitu algoritma Kalman Filter waktu diskrit (*Discrete-Time Kalman Filter*). Oleh karena itu, model gerak longitudinal pesawat terbang didiskritisasi dengan menggunakan metode Beda Hingga Maju karena diprediksi satu langkah ke depan yaitu :

$$\dot{x} = \frac{dx}{dt} \approx \frac{x_{k+1} - x_k}{\Delta t}$$

## (d) Tahap Prediksi

Pada tahap prediksi ini menghitung kovarian error dan estimasi pada model sistem.

## (e) Tahap Koreksi

Pada tahap koreksi ini menghitung Kalman Gain, kovarian error pada model pengukuran kemudian diperoleh hasil estimasi. Pada tahap ini, perhitungan dikatakan bagus jika nilai kovarian errornya semakin kecil. Dan juga perhitungan dikatakan bagus jika nilai errornya semakin kecil pula.

### 3.3.2 Implementasi Fuzzy Kalman Filter

Metode Fuzzy Kalman Filter yang digunakan pada sistem model linear dalam waktu diskrit. Adapun langkah-langkah yang dilakukan untuk estimasi gerak pesawat terbang dengan menggunakan algoritma Fuzzy Kalman Filter adalah sebagai berikut :

## (a) Menentukan Model Sistem dan Model Pengukuran

Berdasarkan persamaan 2.88 - 2.89 diperoleh model sistem dan model pengukuran sebagai berikut:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + G_k w_k \\ z_k &= H_k x_k + v_k \end{aligned}$$

$$w_k \sim N(0, Q_k) \text{ dan } v_k \sim N(0, R_k)$$

dengan  $x = [u, w, q, \theta]^T$ .

(b) Pendiskritan

Penerapan metode Fuzzy Kalman Filter yang digunakan sama dengan algoritma Kalman Filter yaitu dalam waktu diskrit. Oleh karena itu, model gerak longitudinal pesawat terbang didiskritisasi dengan menggunakan metode Beda Hingga Maju karena diprediksi satu langkah ke depan yaitu :

$$\dot{x} = \frac{dx}{dt} \approx \frac{x_{k+1} - x_k}{\Delta t}$$

(c) Proses Fuzzifikasi

Pada tahap ini merupakan proses fuzzifikasi untuk menentukan fungsi keanggotaan dari masing-masing variabel Fuzzy. Fungsi keanggotaan yang diterapkan menggunakan fungsi keanggotaan linear yaitu pada saat maksimum dan minimum.

- Jika variabel minimum, maka fungsi keanggotaannya yaitu

$$\mu_{x_1} = \frac{(x - x^-)}{(x^+ - x^-)}$$

- Jika variabel maksimum, maka fungsi keanggotaannya yaitu

$$\mu_{x_2} = \frac{(x^+ - x)}{(x^+ - x^-)}$$

(d) Menentukan Aturan Dasar Logika Fuzzy

Pada tahap ini, aturan dasar logika Fuzzy diperoleh dari kombinasi dari variabel-variabel minimum dan variabel-variabel maksimum.

(e) Penerapan Kalman Filter

Pada tahap ini dilakukan estimasi dengan metode Kalman Filter waktu diskrit dengan catatan model sistem dan model pengukuran didapatkan melalui proses fuzzifikasi dan aturan dasar logika Fuzzy.

(f) Proses Defuzzifikasi

Pada tahap ini merupakan proses filter untuk memperoleh hasil estimasi yang terbaik. Rumus bobot yang diperoleh berdasarkan himpunan-himpunan Fuzzy yang terbentuk dari kombinasi fungsi keanggotaan dari variabel-variabel Fuzzy sesuai aturan dasar. Setiap himpunan Fuzzy tersebut dikalikan dengan hasil estimasi pada tahap koreksi sesuai aturan. Sehingga rumus bobot tersebut diperoleh dari jumlah dari setiap himpunan Fuzzy yang dikalikan dengan hasil estimasi masing-masing kemudian dibagi dengan jumlah dari himpunan-himpunan Fuzzy.

$$x_{k+1} = \frac{\rho^1 x_{k+1}^1 + \rho^2 x_{k+1}^2 + \dots + \rho^n x_{k+1}^n}{\rho^1 + \rho^2 + \dots + \rho^n}$$

dimana  $i = 1, 2, \dots, n$ .

### 3.4 Simulasi dan Pembahasan

Pada tahap ini dilakukan penerapan model gerak pesawat terbang dengan algoritma Kalman Filter dan Fuzzy Kalman Filter. Selanjutnya dilakukan simulasi dengan menggunakan *software* MATLAB untuk mengetahui hasil estimasi dari metode Kalman Filter dan Fuzzy Kalman Filter. Selain itu, akan dilakukan juga *optimal smoothing*.

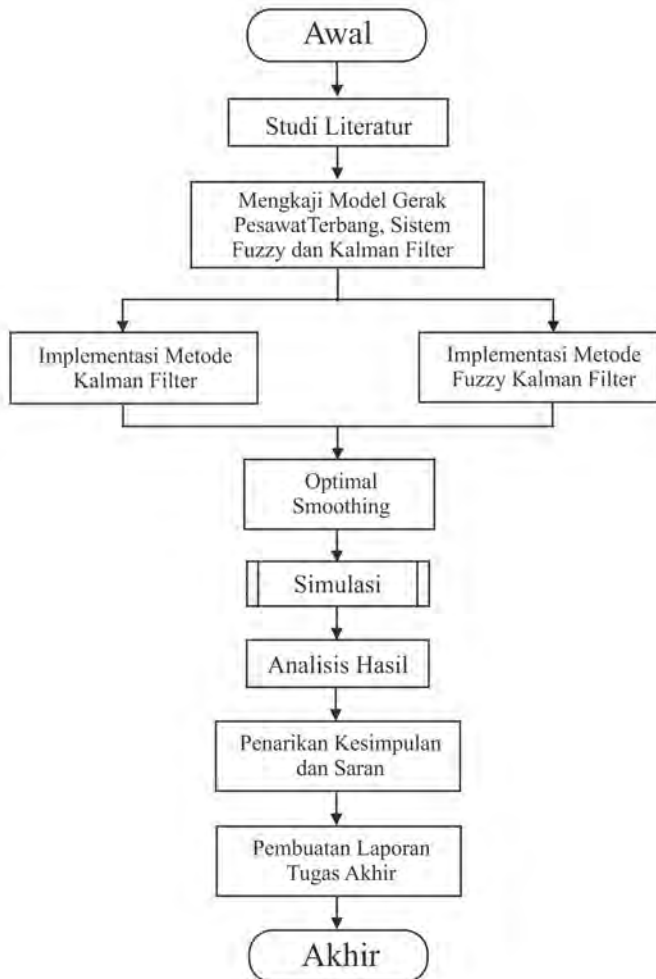
### 3.5 Penarikan Kesimpulan dan Saran

Pada tahap ini dilakukan penarikan kesimpulan berdasarkan hasil simulasi dan pembahasan pada tahap

sebelumnya. Selanjutnya dari hasil kesimpulan-kesimpulan yang terjadi diberikan saran untuk penelitian selanjutnya.

### **3.6 Pembuatan Laporan Tugas Akhir**

Pada tahap ini dilakukan pembuatan laporan Tugas Akhir setelah melakukan simulasi dan menjawab permasalahan yang ada pada topik Tugas Akhir.



Gambar 3.1: Diagram Alur Penelitian

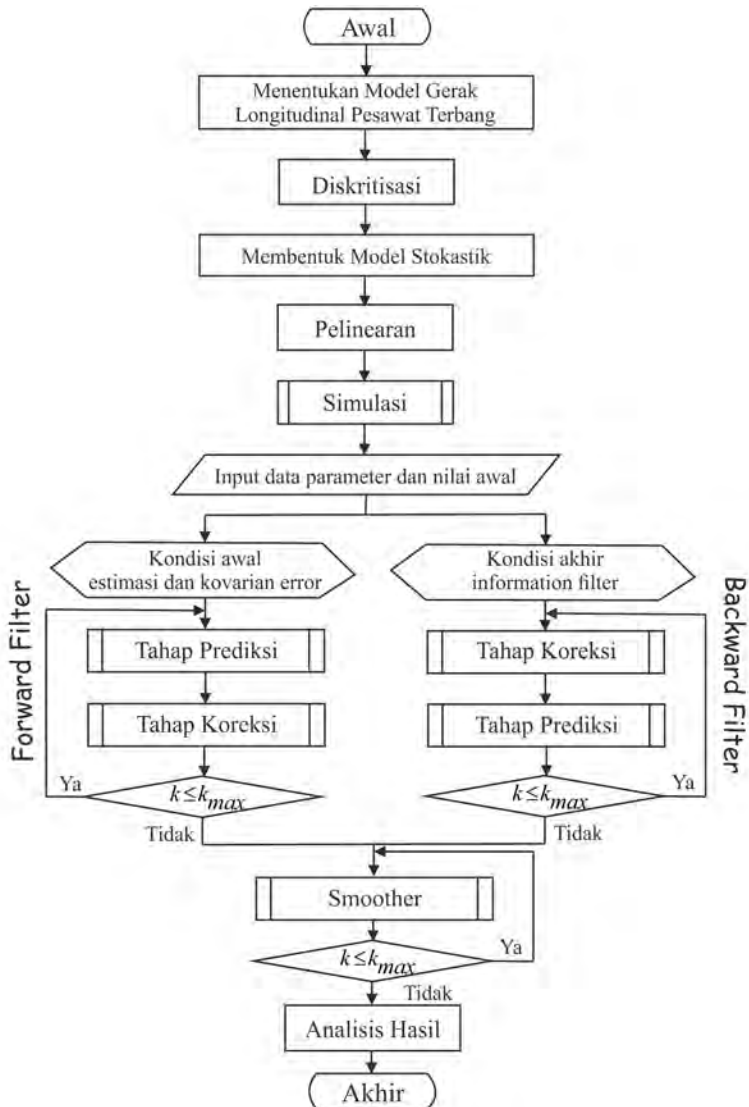


Gambar 3.2: Diagram Alur Kalman Filter

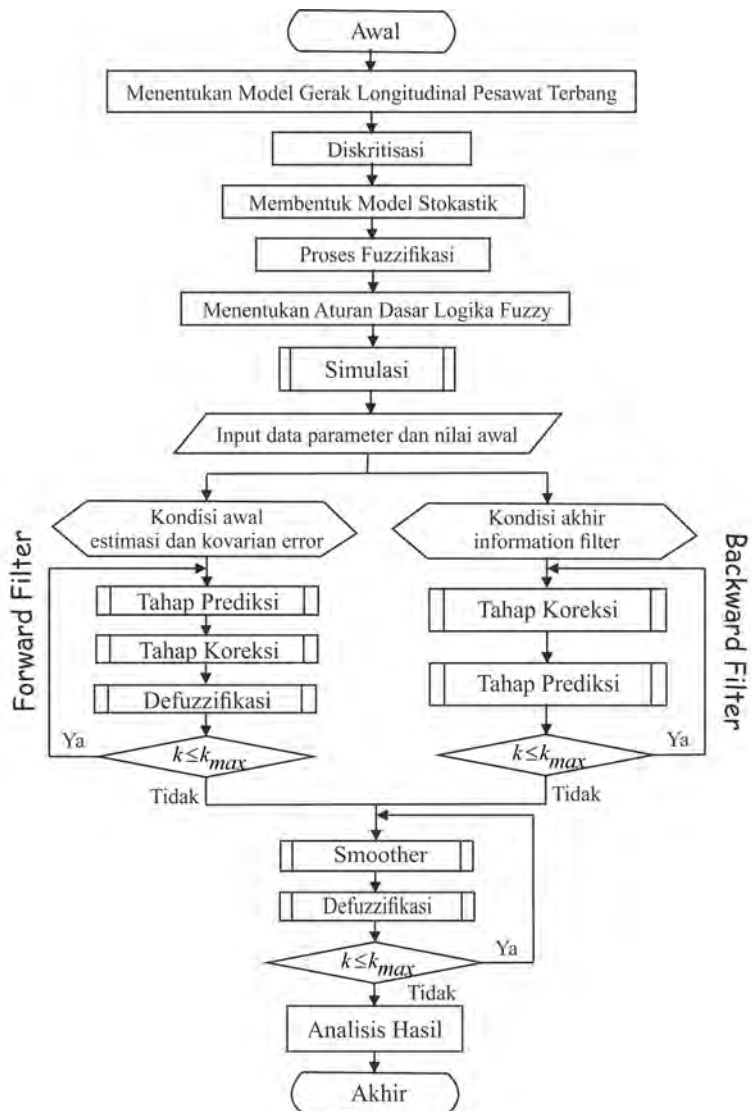




Gambar 3.3: Diagram Alur Fuzzy Kalman Filter



Gambar 3.4: Diagram Alur Optimal Smoothing Kalman Filter



Gambar 3.5: Diagram Alur Optimal Smoothing Fuzzy Kalman Filter

*"Halaman ini sengaja dikosongkan."*

## BAB IV HASIL DAN PEMBAHASAN

Pada bab ini dibahas mengenai estimasi optimal pada gerak longitudinal pesawat terbang. Pembahasan diawali dengan pembentukan model ruang keadaan (*state space*) waktu diskrit stokastik. Selanjutnya dilakukan proses estimasi dengan algoritma Kalman Filter. Dan juga dilakukan pula proses estimasi dengan algoritma Fuzzy Kalman Filter. Metode Fuzzy Kalman Filter dilakukan dengan melalui proses fuzzifikasi, aturan dasar logika Fuzzy dan defuzzifikasi. Setelah itu, melakukan simulasi dengan *software* MATLAB untuk memperoleh tingkat keakurasian dari kedua algoritma dan menganalisis hasil simulasi.

### 4.1 Persamaan Gerak Longitudinal Pesawat Terbang

Gerak longitudinal pesawat terbang terjadi akibat kerja dari sayap *elevator*. Sayap *elevator* merupakan sayap belakang pada pesawat terbang dimana prinsip kerjanya yaitu pada sayap kanan dan kiri (*elevator*) akan naik atau turun (*pitching*). Gerak longitudinal pesawat terbang dipengaruhi oleh kecepatan translasi ke depan, kecepatan translasi ke atas, kecepatan sudut *pitch* dan sudut *pitch*. Persamaan gerak longitudinal yang diperoleh pada persamaan 2.86 yaitu:

$$\begin{aligned}
 \dot{u} &= X_u u + X_w w - wq - g\theta + X_{\delta_e} \delta_e + X_{\delta_T} \delta_T \\
 \dot{w} &= Z_u u + Z_w w + U_0 q + q u + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T \\
 \dot{q} &= (M_u + M_{\dot{w}} Z_u) u + (M_w + M_{\dot{w}} Z_w) w + (M_q + M_{\dot{w}} U_0) q \\
 &\quad + M_{\dot{w}} q u + (M_{\delta_e} + M_{\dot{w}} Z_{\delta_e}) \delta_e + (M_{\delta_T} + M_{\dot{w}} Z_{\delta_T}) \delta_T \\
 \dot{\theta} &= q
 \end{aligned} \tag{4.1}$$

Dengan memisalkan :

$$\begin{aligned}\check{M}_u &= M_u + M_{\dot{w}}Z_u & \check{M}_{\delta_e} &= M_{\delta_e} + M_{\dot{w}}Z_{\delta_e} \\ \check{M}_w &= M_w + M_{\dot{w}}Z_w & \check{M}_{\delta_T} &= M_{\delta_T} + M_{\dot{w}}Z_{\delta_T} \\ \check{M}_q &= M_q + M_{\dot{w}}U_0\end{aligned}$$

Sehingga diperoleh persamaan ruang keadaan waktu kontinu yaitu

$$\begin{aligned}\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} &= \begin{bmatrix} X_u & X_w & -w & -g \\ Z_u + q & Z_w & U_0 & 0 \\ \check{M}_u & \check{M}_w & \check{M}_q + M_{\dot{w}}u & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \\ &+ \begin{bmatrix} X_{\delta_e} & X_{\delta_T} \\ Z_{\delta_e} & Z_{\delta_T} \\ \check{M}_{\delta_e} & \check{M}_{\delta_T} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_T \end{bmatrix} \quad (4.2)\end{aligned}$$

dan outputnya berdasarkan penelitian yang dirujuk, variabel yang dapat diamati yaitu  $w, q$ , sehingga diperoleh

$$z_k = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \quad (4.3)$$

dimana

- $u$  : Kecepatan translasi ke depan
- $w$  : Kecepatan translasi ke atas
- $q$  : Kecepatan sudut *pitch*
- $\theta$  : Sudut *pitch*

#### 4.1.1 Diskritisasi

Persamaan gerak longitudinal pesawat terbang tersebut merupakan model sistem dinamik deterministik waktu kontinu. Persamaan gerak longitudinal pesawat terbang tersebut diubah menjadi bentuk model sistem dinamik waktu diskrit. Berdasarkan persamaan 4.1 untuk memperoleh sistem persamaan waktu diskrit dengan cara melakukan proses pendiskritan dengan menggunakan metode beda hingga maju, sebagai berikut :

$$\begin{aligned} \dot{u} = \frac{du}{dt} &= \frac{u_{k+1} - u_k}{\Delta t} \quad ; \quad \dot{w} = \frac{dw}{dt} = \frac{w_{k+1} - w_k}{\Delta t} \\ \dot{q} = \frac{dq}{dt} &= \frac{q_{k+1} - q_k}{\Delta t} \quad ; \quad \dot{\theta} = \frac{d\theta}{dt} = \frac{\theta_{k+1} - \theta_k}{\Delta t} \end{aligned}$$

Sehingga persamaan 4.1 menjadi

$$\begin{aligned} \frac{u_{k+1} - u_k}{\Delta t} &= X_u u_k + X_w w_k - w_k q_k - g \theta_k + X_{\delta_e} \delta_{e_k} + X_{\delta_T} \delta_{T_k} \\ \frac{w_{k+1} - w_k}{\Delta t} &= Z_u u_k + Z_w w_k + U_0 q_k + q_k u_k + Z_{\delta_e} \delta_{e_k} + Z_{\delta_T} \delta_{T_k} \\ \frac{q_{k+1} - q_k}{\Delta t} &= \check{M}_u u_k + \check{M}_w w_k + \check{M}_q q_k + M_{\dot{w}} q_k u_k + \check{M}_{\delta_e} \delta_{e_k} + \check{M}_{\delta_T} \delta_{T_k} \\ \frac{\theta_{k+1} - \theta_k}{\Delta t} &= q_k \end{aligned} \tag{4.4}$$

Persamaan 4.4 dioperasikan sehingga bentuknya menjadi

$$\begin{aligned} u_{k+1} - u_k &= X_u \Delta t u_k + X_w \Delta t w_k - \Delta t w_k q_k - g \Delta t \theta_k \\ &\quad + X_{\delta_e} \Delta t \delta_{e_k} + X_{\delta_T} \Delta t \delta_{T_k} \\ w_{k+1} - w_k &= Z_u \Delta t u_k + Z_w \Delta t w_k + U_0 \Delta t q_k + \Delta t q_k u_k \\ &\quad + Z_{\delta_e} \Delta t \delta_{e_k} + Z_{\delta_T} \Delta t \delta_{T_k} \\ q_{k+1} - q_k &= \check{M}_u \Delta t u_k + \check{M}_w \Delta t w_k + \check{M}_q \Delta t q_k + M_{\dot{w}} \Delta t \\ &\quad q_k u_k + \check{M}_{\delta_e} \Delta t \delta_{e_k} + \check{M}_{\delta_T} \Delta t \delta_{T_k} \\ \theta_{k+1} - \theta_k &= \Delta t q_k \end{aligned}$$

dan dapat ditulis menjadi

$$\begin{aligned}
u_{k+1} &= (X_u \Delta t + 1) u_k + X_w \Delta t w_k - \Delta t w_k q_k - g \Delta t \theta_k \\
&\quad + X_{\delta_e} \Delta t \delta_{e_k} + X_{\delta_T} \Delta t \delta_{T_k} \\
w_{k+1} &= Z_u \Delta t u_k + (Z_w \Delta t + 1) w_k + U_0 \Delta t q_k + \Delta t q_k u_k \\
&\quad + Z_{\delta_e} \Delta t \delta_{e_k} + Z_{\delta_T} \Delta t \delta_{T_k} \\
q_{k+1} &= \check{M}_u \Delta t u_k + \check{M}_w \Delta t w_k + (\check{M}_q \Delta t + 1) q_k + M_{\dot{w}} \Delta t \\
&\quad q_k u_k + \check{M}_{\delta_e} \Delta t \delta_{e_k} + \check{M}_{\delta_T} \Delta t \delta_{T_k} \\
\theta_{k+1} &= \Delta t q_k + \theta_k
\end{aligned} \tag{4.5}$$

Pada persamaan 4.5 merupakan model gerak longitudinal pesawat terbang waktu diskrit. Sehingga sistem model gerak longitudinal pesawat terbang waktu diskrit dapat disajikan dalam bentuk persamaan ruang keadaan (*state space*) yaitu

$$\begin{bmatrix} u_{k+1} \\ w_{k+1} \\ q_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w_k & -g \Delta t \\ (Z_u + q_k) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}} u_k) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

$$\begin{bmatrix} u_k \\ w_k \\ q_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \Delta t & X_{\delta_T} \Delta t \\ Z_{\delta_e} \Delta t & Z_{\delta_T} \Delta t \\ \check{M}_{\delta_e} \Delta t & \check{M}_{\delta_T} \Delta t \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{e_k} \\ \delta_{T_k} \end{bmatrix} \tag{4.6}$$

dan outpunya juga memuat noise pengukuran, sehingga diperoleh

$$z_k = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ w_k \\ q_k \\ \theta_k \end{bmatrix} \tag{4.7}$$



#### 4.1.2 Pembentukan Sistem Diskrit Stokastik

Model gerak longitudinal pesawat terbang pada persamaan 4.5 merupakan sistem deterministik. Model sistem deterministik dari gerak longitudinal pesawat terbang tersebut mengabaikan adanya suatu noise. Noise-noise tersebut dapat terjadi pada model sistem seperti kesalahan dalam memodelkan dan juga noise dapat terjadi pada model pengukuran seperti kesalahan dalam pengukuran. Meskipun noise berukuran sangat kecil namun perlu diperhitungkan adanya suatu noise. Sistem persamaan gerak longitudinal pesawat terbang yang sebenarnya yaitu, maka persamaan 4.5 menjadi

$$\begin{aligned}
 u_{k+1} &= (X_u \Delta t + 1) u_k + X_w \Delta t w_k - \Delta t w_k q_k - g \Delta t \theta_k \\
 &\quad + X_{\delta_e} \Delta t \delta_{e_k} + X_{\delta_T} \Delta t \delta_{T_k} + \xi_{1_k} \\
 w_{k+1} &= Z_u \Delta t u_k + (Z_w \Delta t + 1) w_k + U_0 \Delta t q_k + \Delta t q_k u_k \\
 &\quad + Z_{\delta_e} \Delta t \delta_{e_k} + Z_{\delta_T} \Delta t \delta_{T_k} + \xi_{1_k} \\
 q_{k+1} &= \check{M}_u \Delta t u_k + \check{M}_w \Delta t w_k + (\check{M}_q \Delta t + 1) q_k + \check{M}_{\dot{w}} \Delta t \\
 &\quad q_k u_k + \check{M}_{\delta_e} \Delta t \delta_{e_k} + \check{M}_{\delta_T} \Delta t \delta_{T_k} + \xi_{2_k} \\
 \theta_{k+1} &= \Delta t q_k + \theta_k + \xi_{3_k}
 \end{aligned} \tag{4.8}$$

Bentuk persamaan ruang keadaan dari sistem dinamik stokastik diskrit [8] yaitu

$$\begin{aligned}
 x_{k+1} &= A_k x_k + B_k u_k + G_k w_k \\
 z_k &= H_k x_k + v_k
 \end{aligned}$$

dimana  $w_k$  adalah noise sistem dan  $v_k$  adalah noise pengukuran.

Atau dapat ditulis dalam bentuk penulisan lainnya yaitu

$$\begin{aligned}
 x_{k+1} &= A_k x_k + B_k u_k + G_k \xi_k \\
 z_k &= H_k x_k + \eta_k
 \end{aligned}$$

Dan juga untuk  $\xi_k$  adalah noise sistem dan  $\eta_k$  adalah noise pengukuran. Persamaan 4.8 dibentuk menjadi persamaan ruang keadaan dengan sistem dinamik stokastik waktu diskrit dari model gerak longitudinal pesawat terbang, sebagai berikut :

$$\begin{bmatrix} u_{k+1} \\ w_{k+1} \\ q_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w_k & -g \Delta t \\ (Z_u + q_k) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}} u_k) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

$$\begin{bmatrix} u_k \\ w_k \\ q_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \Delta t & X_{\delta_T} \Delta t \\ Z_{\delta_e} \Delta t & Z_{\delta_T} \Delta t \\ \check{M}_{\delta_e} \Delta t & \check{M}_{\delta_T} \Delta t \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{e_k} \\ \delta_{T_k} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_{1_k} \\ \xi_{2_k} \\ \xi_{3_k} \end{bmatrix} \quad (4.9)$$

dan keluarannya

$$z_k = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ w_k \\ q_k \\ \theta_k \end{bmatrix} + \eta_k \quad (4.10)$$

Sehingga diperoleh

$$A = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w_k & -g \Delta t \\ (Z_u + q_k) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}} u_k) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} X_{\delta_e} \Delta t & X_{\delta_T} \Delta t \\ Z_{\delta_e} \Delta t & Z_{\delta_T} \Delta t \\ \check{M}_{\delta_e} \Delta t & \check{M}_{\delta_T} \Delta t \\ 0 & 0 \end{bmatrix} ; G = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

dengan

- $u_k$  : Kecepatan translasi ke depan
- $w_k$  : Kecepatan translasi ke atas
- $q_k$  : Kecepatan sudut *pitch*
- $\theta_k$  : Sudut *pitch*

Berdasarkan data hasil penelitian oleh Sukandi (2010) diperoleh nilai parameter-parameter, sebagai berikut :

Tabel 4.1: Nilai parameter gerak longitudinal pesawat terbang

Parameter	Nilai	Parameter	Nilai
$X_u$	0.0002	$Z_{\delta_e}$	-5.46
$X_w$	0.039	$Z_{\delta_T}$	$-1.5 \times 10^{-7}$
$X_{\delta_e}$	0.44	$\check{M}_u$	$8.8 \times 10^{-5}$
$X_{\delta_T}$	$3.434 \times 10^{-6}$	$\check{M}_w$	-0.002873
$Z_u$	-0.07	$\check{M}_q$	-0.439
$Z_w$	-0.317	$\check{M}_{\delta_e}$	-1.1578
$U_0$	27.78	$\check{M}_{\delta_T}$	$6.701 \times 10^{-7}$
$M_{\dot{w}}$	-0.0004		

## 4.2 Implementasi Kalman Filter

Metode Kalman Filter yang digunakan untuk mengestimasi gerak longitudinal pesawat terbang membutuhkan sistem diskrit yang linear. Model gerak longitudinal pesawat terbang pada persamaan 4.8 merupakan sistem nonlinear. Bentuk model sistem dan model pengukuran :

$$\begin{aligned}
 x_{k+1} &= A_k x_k + B_k u_k + G_k \xi_k \\
 z_k &= H_k x_k + \eta_k
 \end{aligned}$$

dengan  $x_k = [u, w, q, \theta]^T$  Sehingga untuk mendapatkan sistem yang linear dengan menggunakan metode Jacobian. Dimana

metode Jacobian ini untuk menentukan sistem persamaan keadaan. Pada persamaan 4.5 dengan memisalkan, yaitu:

$$\begin{aligned} u_{k+1} &= f_1(u_k, w_k, q_k, \theta_k, \delta_{e_k}, \delta_{e_T}) \\ w_{k+1} &= f_2(u_k, w_k, q_k, \theta_k, \delta_{e_k}, \delta_{e_T}) \\ q_{k+1} &= f_3(u_k, w_k, q_k, \theta_k, \delta_{e_k}, \delta_{e_T}) \\ \theta_{k+1} &= f_4(u_k, w_k, q_k, \theta_k, \delta_{e_k}, \delta_{e_T}) \end{aligned}$$

Sehingga diperoleh sistem keadaan dengan metode Jacobian yaitu

$$J_x = \begin{bmatrix} \frac{\partial f_1}{\partial u_k} & \frac{\partial f_1}{\partial w_k} & \frac{\partial f_1}{\partial q_k} & \frac{\partial f_1}{\partial \theta_k} \\ \frac{\partial f_2}{\partial u_k} & \frac{\partial f_2}{\partial w_k} & \frac{\partial f_2}{\partial q_k} & \frac{\partial f_2}{\partial \theta_k} \\ \frac{\partial f_3}{\partial u_k} & \frac{\partial f_3}{\partial w_k} & \frac{\partial f_3}{\partial q_k} & \frac{\partial f_3}{\partial \theta_k} \\ \frac{\partial f_4}{\partial u_k} & \frac{\partial f_4}{\partial w_k} & \frac{\partial f_4}{\partial q_k} & \frac{\partial f_4}{\partial \theta_k} \end{bmatrix}; J_u = \begin{bmatrix} \frac{\partial f_1}{\partial \delta_{e_k}} & \frac{\partial f_1}{\partial \delta_{T_k}} \\ \frac{\partial f_2}{\partial \delta_{e_k}} & \frac{\partial f_2}{\partial \delta_{T_k}} \\ \frac{\partial f_3}{\partial \delta_{e_k}} & \frac{\partial f_3}{\partial \delta_{T_k}} \\ \frac{\partial f_4}{\partial \delta_{e_k}} & \frac{\partial f_4}{\partial \delta_{T_k}} \end{bmatrix}$$

sehingga diperoleh

$$A = \begin{bmatrix} X_u \Delta t + 1 & (X_w - q_k) \Delta t & -\Delta t w_k & -g \Delta t \\ (Z_u + q_k) \Delta t & Z_w \Delta t + 1 & (U_0 + u_k) \Delta t & 0 \\ (\check{M}_u + \check{M}_{\dot{w}} q_k) \Delta t & \check{M}_w \Delta t & (\check{M}_q + \check{M}_{\dot{w}} u_k) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

Langkah awal dari algoritma Kalman Filter membutuhkan nilai awal dari variabel-variabel gerak longitudinal pesawat terbang. Dimana variabel gerak longitudinal pesawat terbang meliputi yaitu kecepatan translasi ke depan ( $u$ ), kecepatan translasi ke atas ( $w$ ), kecepatan sudut *pitch* ( $q$ ) dan sudut *pitch* ( $\theta$ ).

$$x = [u, w, q, \theta]^T$$

Dan diberikan nilai awal yaitu

$$x_0 = [u_0, w_0, q_0, \theta_0]^T$$

dimana kecepatan awal translasi ke depan ( $u_0$ ), kecepatan awal translasi ke atas ( $w_0$ ), kecepatan sudut *pitch* awal ( $q_0$ ) dan sudut *pitch* awal ( $\theta_0$ ).

- model sistem dan model pengukuran

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + G\xi_k \\z_k &= Hx_k + \eta_k\end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

Dengan metode jacobian sehingga diperoleh sistem diskrit linear.

Diperoleh model sistem :

$$\begin{bmatrix} u_{k+1} \\ w_{k+1} \\ q_{k+1} \\ \theta_{k+1} \end{bmatrix} =$$

$$\begin{bmatrix} X_u \Delta t + 1 & (X_w - q_k) \Delta t & -\Delta t w_k & -g \Delta t \\ (Z_u + q_k) \Delta t & Z_w \Delta t + 1 & (U_0 + u_k) \Delta t & 0 \\ (\check{M}_u + \check{M}_{\dot{w}} q_k) \Delta t & \check{M}_w \Delta t & (\check{M}_q + \check{M}_{\dot{w}} u_k) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

$$\begin{bmatrix} u_k \\ w_k \\ q_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \Delta t & X_{\delta_T} \Delta t \\ Z_{\delta_e} \Delta t & Z_{\delta_T} \Delta t \\ \check{M}_{\delta_e} \Delta t & \check{M}_{\delta_T} \Delta t \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{e_k} \\ \delta_{T_k} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_{1_k} \\ \xi_{2_k} \\ \xi_{3_k} \end{bmatrix}$$

dan model pengukuran :

$$z_k = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ w_k \\ q_k \\ \theta_k \end{bmatrix} + \eta_k$$

- inisialisasi

Pada langkah ini diberikan inisialisasi awal untuk estimasi awal ( $\hat{x}_0$ ) dan kovarian  $x$  ( $P_0$ ) yaitu

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{\bar{k}+1} = A_k P_k A_k^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{\bar{k}+1} = A_k \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1} = P_{k+1} H_{k+1}^T (H_{k+1} P_{k+1} H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1} = (I - K_{k+1} H_{k+1}) P_{\bar{k}+1} \\ \text{estimasi} &: \hat{x}_{k+1} = \hat{x}_{\bar{k}+1} + K_{k+1} (z_{k+1} - H_{k+1} \hat{x}_{\bar{k}+1}) \end{aligned}$$

Dimana  $z_k$  merupakan data pengukuran yang bersifat random. Data yang diukur yaitu kecepatan translasi ke atas ( $w$ ) dan kecepatan sudut *pitch* ( $q$ ). Kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

### 4.3 Implementasi Fuzzy Kalman Filter

Metode Fuzzy Kalman Filter merupakan metode kombinasi dari Logika Fuzzy dan Kalman Filter. Metode Fuzzy Kalman Filter yang digunakan melalui tahapan-tahapan dari Logika Fuzzy dan selanjutnya di terapkan pada algoritma Kalman Filter. Berikut ini adalah langkah-langkah dari metode Fuzzy Kalman Filter, sebagai berikut :

#### 4.3.1 Fuzzifikasi

Bentuk sistem persamaan yaitu :

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + G\xi_k \\ z_k &= Hx_k + \eta_k \end{aligned}$$

Berdasarkan persamaan 4.9 merupakan persamaan nonlinear. Dan diperoleh bahwa dalam matriks  $A$  termuat variabel-variabel gerak longitudinal pesawat terbang yaitu kecepatan translasi ke depan ( $u$ ), kecepatan translasi ke atas ( $w$ ) dan kecepatan sudut *pitch* ( $q$ ). Dengan proses fuzzifikasi, variabel-variabel gerak longitudinal pesawat terbang yang termuat dalam sistem matriks  $A$  ditentukan pada interval masing-masing, sebagai berikut :

$$\begin{aligned} u &\in [u^-, u^+] \\ w &\in [w^-, w^+] \\ q &\in [q^-, q^+] \end{aligned}$$

artinya :

- $u \in [u^-, u^+]$  yaitu  
 $u^-$  : kecepatan translasi ke depan minimum  
 $u^+$  : kecepatan translasi ke depan maksimum
- $w \in [w^-, w^+]$  yaitu  
 $w^-$  : kecepatan translasi ke atas minimum  
 $w^+$  : kecepatan translasi ke atas maksimum
- $q \in [q^-, q^+]$  yaitu  
 $q^-$  : kecepatan sudut *pitch* minimum  
 $q^+$  : kecepatan sudut *pitch* maksimum

Sehingga fungsi keanggotaannya diperoleh sebagai berikut :

1. Untuk kecepatan translasi ke depan ( $u$ )

- a. Jika  $u$  minimum :

$$\mu_{u_{min}} = \frac{u - u^-}{u^+ - u^-}$$

- b. Jika  $u$  maksimum :

$$\mu_{u_{max}} = \frac{u^+ - u}{u^+ - u^-}$$

2. Untuk kecepatan translasi ke atas ( $w$ )

- a. Jika  $w$  minimum :

$$\mu_{w_{min}} = \frac{w - w^-}{w^+ - w^-}$$

- b. Jika  $w$  maksimum :

$$\mu_{w_{max}} = \frac{w^+ - w}{w^+ - w^-}$$



3. Untuk kecepatan sudut *pitch* ( $q$ )

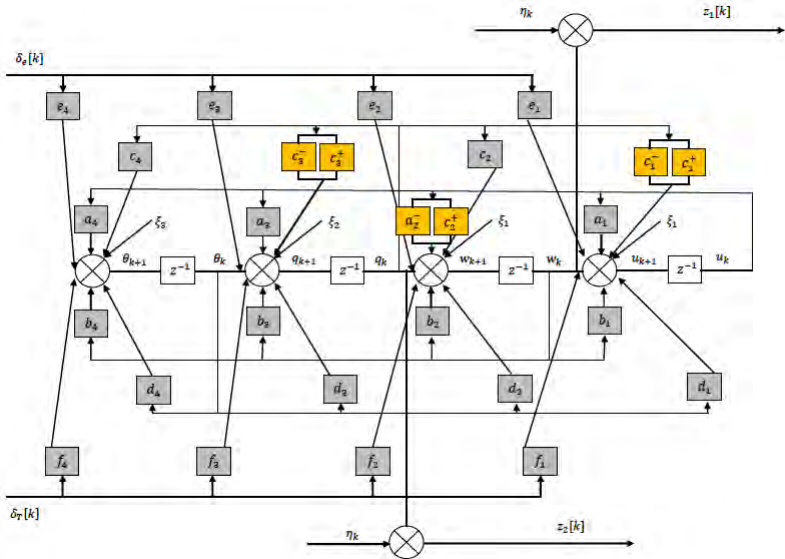
a. Jika  $q$  minimum :

$$\mu_{q_{min}} = \frac{q - q^-}{q^+ - q^-}$$

b. Jika  $q$  maksimum :

$$\mu_{q_{max}} = \frac{q^+ - q}{q^+ - q^-}$$

Berikut ini adalah diagram blok dari sistem Fuzzy untuk model gerak longitudinal pesawat terbang:



Gambar 4.1: Diagram Blok Fuzzy untuk Model Gerak Longitudinal Pesawat Terbang

#### 4.3.2 Aturan Dasar Logika Fuzzy

Variabel-variabel gerak longitudinal pesawat terbang terdapat tiga variabel yang difuzzikan yaitu  $u, w, q$ . Berdasarkan fungsi keanggotaan yang diperoleh sebelumnya sehingga terdapat  $2^n$  aturan. Dimana  $n$  merupakan banyaknya variabel yang difuzzikan. Sehingga diperoleh 8 aturan. Sesuai aturan dasar logika Fuzzy, yaitu :

*rule i* : IF ... THEN ...

maka :

- rule 1 : IF  $u$  is  $u^-$  and  $w$  is  $w^-$  and  $q$  is  $q^-$  THEN  $A^1$
- rule 2 : IF  $u$  is  $u^-$  and  $w$  is  $w^-$  and  $q$  is  $q^+$  THEN  $A^2$
- rule 3 : IF  $u$  is  $u^-$  and  $w$  is  $w^+$  and  $q$  is  $q^-$  THEN  $A^3$
- rule 4 : IF  $u$  is  $u^-$  and  $w$  is  $w^+$  and  $q$  is  $q^+$  THEN  $A^4$
- rule 5 : IF  $u$  is  $u^+$  and  $w$  is  $w^-$  and  $q$  is  $q^-$  THEN  $A^5$
- rule 6 : IF  $u$  is  $u^+$  and  $w$  is  $w^-$  and  $q$  is  $q^+$  THEN  $A^6$
- rule 7 : IF  $u$  is  $u^+$  and  $w$  is  $w^+$  and  $q$  is  $q^-$  THEN  $A^7$
- rule 8 : IF  $u$  is  $u^+$  and  $w$  is  $w^+$  and  $q$  is  $q^+$  THEN  $A^8$

#### 4.3.3 Algoritma Fuzzy Kalman Filter

Algoritma Fuzzy Kalman Filter merupakan penerapan dari algoritma Kalman Filter dengan proses fuzzifikasi sesuai aturan dasar logika Fuzzy. Berdasarkan proses fuzzifikasi dan aturan dasar logika Fuzzy, sistem gerak longitudinal pesawat terbang diperoleh terdapat 8 aturan yang akan diterapkan pada algoritma Fuzzy Kalman Filter. Berikut adalah algoritma Fuzzy Kalman Filter :

1. Rule ke-1 :

Diperoleh bahwa

$$A^1 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^- & -g \Delta t \\ (Z_u + q^-) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}u^-}) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned}x_{k+1}^1 &= A_k^1 x_k + B_k u_k + G_k \xi_k \\ z_k &= H_k x_k + \eta_k\end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned}\text{kovarian error} &: P_{k+1}^1 = A_k^1 P_k (A_k^1)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^1 = A_k^1 \hat{x}_k + B_k u_k\end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} & : K_{k+1}^1 = P_{k+1}^1 H_{k+1}^T (H_{k+1} P_{k+1}^1 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} & : P_{k+1}^1 = (I - K_{k+1}^1 H_{k+1}) P_{k+1}^1 \\ \text{estimasi} & : \hat{x}_{k+1}^1 = \hat{x}_{k+1}^1 + K_{k+1}^1 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^1) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

2. Rule ke-2 :

Diperoleh bahwa

$$A^2 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^- & -g \Delta t \\ (Z_u + q^+) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}} u^-) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned} x_{k+1}^2 & = A_k^2 x_k + B_k u_k + G_k \xi_k \\ z_k & = H_k x_k + \eta_k \end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^2 = A_k^2 P_k (A_k^2)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^2 = A_k^2 \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^2 = P_{k+1}^2 H_{k+1}^T (H_{k+1} P_{k+1}^2 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^2 = (I - K_{k+1}^2 H_{k+1}) P_{k+1}^2 \\ \text{estimasi} &: \hat{x}_{k+1}^2 = \hat{x}_{k+1}^2 + K_{k+1}^2 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^2) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

## 3. Rule ke-3 :

Diperoleh bahwa

$$A^3 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^+ & -g \Delta t \\ (Z_u + q^-) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + \check{M}_w u^-) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned} x_{k+1}^3 &= A_k^3 x_k + B_k u_k + G_k \xi_k \\ z_k &= H_k x_k + \eta_k \end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^3 = A_k^3 P_k (A_k^3)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^3 = A_k^3 \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^3 = P_{k+1}^3 H_{k+1}^T (H_{k+1} P_{k+1}^3 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^3 = (I - K_{k+1}^3 H_{k+1}) P_{k+1}^3 \\ \text{estimasi} &: \hat{x}_{k+1}^3 = \hat{x}_{k+1}^3 + K_{k+1}^3 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^3) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

#### 4. Rule ke-4 :

Diperoleh bahwa

$$A^4 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^+ & -g \Delta t \\ (Z_u + q^+) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_w u^-) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned} x_{k+1}^4 &= A_k^4 x_k + B_k u_k + G_k \xi_k \\ z_k &= H_k x_k + \eta_k \end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^4 = A_k^4 P_k (A_k^4)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^4 = A_k^4 \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^4 = P_{k+1}^4 H_{k+1}^T (H_{k+1} P_{k+1}^4 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^4 = (I - K_{k+1}^4 H_{k+1}) P_{k+1}^4 \\ \text{estimasi} &: \hat{x}_{k+1}^4 = \hat{x}_{k+1}^4 + K_{k+1}^4 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^4) \end{aligned}$$



Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

5. Rule ke-5 :

Diperoleh bahwa

$$A^5 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^- & -g \Delta t \\ (Z_u + q^-) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + \check{M}_w u^+) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned} x_{k+1}^5 &= A_k^5 x_k + B_k u_k + G_k \xi_k \\ z_k &= H_k x_k + \eta_k \end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^5 = A_k^5 P_k (A_k^5)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^5 = A_k^5 \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^5 = P_{k+1}^5 H_{k+1}^T (H_{k+1} P_{k+1}^5 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^5 = (I - K_{k+1}^5 H_{k+1}) P_{k+1}^5 \\ \text{estimasi} &: \hat{x}_{k+1}^5 = \hat{x}_{k+1}^5 + K_{k+1}^5 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^5) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

## 6. Rule ke-6 :

Diperoleh bahwa

$$A^6 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^- & -g \Delta t \\ (Z_u + q^+) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + \check{M}_w u^+) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned}x_{k+1}^6 &= A_k^6 x_k + B_k u_k + G_k \xi_k \\6_k &= H_k x_k + \eta_k\end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned}\text{kovarian error} &: P_{k+1}^6 = A_k^6 P_k (A_k^6)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^6 = A_k^6 \hat{x}_k + B_k u_k\end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^6 = P_{k+1}^6 H_{k+1}^T (H_{k+1} P_{k+1}^6 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^6 = (I - K_{k+1}^6 H_{k+1}) P_{k+1}^6 \\ \text{estimasi} &: \hat{x}_{k+1}^6 = \hat{x}_{k+1}^6 + K_{k+1}^6 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^6) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

#### 7. Rule ke-7 :

Diperoleh bahwa

$$A^7 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^+ & -g \Delta t \\ (Z_u + q^-) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}} u^+) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned} x_{k+1}^3 &= A_k^3 x_k + B_k u_k + G_k \xi_k \\ 7_k &= H_k x_k + \eta_k \end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^7 = A_k^7 P_k (A_k^7)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^7 = A_k^7 \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^7 = P_{k+1}^7 H_{k+1}^T (H_{k+1} P_{k+1}^7 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^7 = (I - K_{k+1}^7 H_{k+1}) P_{k+1}^7 \\ \text{estimasi} &: \hat{x}_{k+1}^7 = \hat{x}_{k+1}^7 + K_{k+1}^7 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^7) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

8. Rule ke-8 :

Diperoleh bahwa

$$A^8 = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w^+ & -g \Delta t \\ (Z_u + q^+) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + \check{M}_{\dot{w}} u^+) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- model sistem dan model pengukuran

$$\begin{aligned} x_{k+1}^8 &= A_k^8 x_k + B_k u_k + G_k \xi_k \\ \delta_k &= H_k x_k + \eta_k \end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- inisialisasi

Diberikan inisialisasi awal, yaitu :

$$\hat{x}_0 = \bar{x}_0, P_0 = P_{x_0}$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_{x_0}$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

- tahap prediksi (*time update*)

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^8 = A_k^8 P_k (A_k^8)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^8 = A_k^8 \hat{x}_k + B_k u_k \end{aligned}$$

Dimana kovarian dari noise sistem ( $Q_k$ ) merupakan matriks diagonal dengan ukuran  $3 \times 3$ .

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

- tahap koreksi (*measurement update*)

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^8 = P_{k+1}^8 H_{k+1}^T (H_{k+1} P_{k+1}^8 H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^8 = (I - K_{k+1}^8 H_{k+1}) P_{k+1}^8 \\ \text{estimasi} &: \hat{x}_{k+1}^8 = \hat{x}_{k+1}^8 + K_{k+1}^8 (z_{k+1} - H_{k+1} \hat{x}_{k+1}^8) \end{aligned}$$

Dimana kovarian dari noise pengukuran ( $R_k$ ) merupakan matriks diagonal dengan ukuran  $2 \times 2$ .

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

#### 4.3.4 Defuzzifikasi

Proses defuzzifikasi merupakan proses filter untuk mendapatkan hasil estimasi secara keseluruhan dengan aturan-aturan yang terbentuk. Setelah masing-masing aturan melalui tahap koreksi, selanjutnya akan diproses secara keseluruhan sesuai 8 aturan tersebut. Hasil estimasi diperoleh sebagai berikut :

$$\hat{x}_{k+1}^i = \begin{bmatrix} \hat{x}_{k+1}^1 \\ \hat{x}_{k+1}^2 \\ \hat{x}_{k+1}^3 \\ \hat{x}_{k+1}^4 \\ \hat{x}_{k+1}^5 \\ \hat{x}_{k+1}^6 \\ \hat{x}_{k+1}^7 \\ \hat{x}_{k+1}^8 \end{bmatrix}$$

Berdasarkan rumus bobot rata-rata, sebagai berikut [11] :

$$\hat{x}_{k+1} = \frac{\rho^1 \hat{x}_{k+1}^1 + \rho^2 \hat{x}_{k+1}^2 + \rho^3 \hat{x}_{k+1}^3 + \dots + \rho^8 \hat{x}_{k+1}^8}{\rho^1 + \rho^2 + \rho^3 + \dots + \rho^8}$$

dimana nilai dari masing-masing  $\rho^1, \rho^2, \rho^3, \rho^4, \rho^5, \rho^6, \rho^7, \rho^8$  ditentukan dari kombinasi fungsi keanggotaan sesuai aturan (*rule*), sebagai berikut :

$$\begin{aligned}\rho^1 &= \mu_{u_{min}} \cdot \mu_{w_{min}} \cdot \mu_{q_{min}} \\ \rho^2 &= \mu_{u_{min}} \cdot \mu_{w_{min}} \cdot \mu_{q_{max}} \\ \rho^3 &= \mu_{u_{min}} \cdot \mu_{w_{max}} \cdot \mu_{q_{min}} \\ \rho^4 &= \mu_{u_{min}} \cdot \mu_{w_{max}} \cdot \mu_{q_{max}} \\ \rho^5 &= \mu_{u_{max}} \cdot \mu_{w_{min}} \cdot \mu_{q_{min}} \\ \rho^6 &= \mu_{u_{max}} \cdot \mu_{w_{min}} \cdot \mu_{q_{max}} \\ \rho^7 &= \mu_{u_{max}} \cdot \mu_{w_{max}} \cdot \mu_{q_{min}} \\ \rho^8 &= \mu_{u_{max}} \cdot \mu_{w_{max}} \cdot \mu_{q_{max}}\end{aligned}$$

Nilai estimasi pada waktu ke-k+1 akan kembali diproses melalui tahap prediksi dan koreksi hingga diperoleh nilai estimasi akhir sesuai waktu yang ditentukan.

#### 4.4 Implementasi *Optimal Smoothing* Kalman Filter

Pada *optimal smoothing* Kalman Filter ini digunakan untuk penghalusan hasil estimasi dari estimasi Kalman Filter. Pada *optimal smoothing* membutuhkan data pengukuran dari Kalman Filter. Dengan melakukan hal yang sama pada implementasi Kalman Filter yaitu perlu dilakukan pelinearan dengan metode Jacobian. Sehingga implementasi *optimal smoothing* Kalman Filter, sebagai berikut:

- model sistem dan model pengukuran

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + G\xi_k \\ z_k &= Hx_k + \eta_k\end{aligned}$$



$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

- Forward Filter

Pada tahap ini, proses estimasi berjalan maju dari awal hingga akhir.

i) inisialisasi:

Pada langkah ini diberikan inisialisasi awal untuk estimasi ( $\hat{x}_0^f$ ) dan kovarian ( $P_0^f$ ) yaitu

$$\hat{x}_0^f = \bar{x}_0, P_0^f = P_0$$

dimana  $\bar{x}_0 = [u_0, w_0, q_0, \theta_0]^T$  dan  $P_0$  merupakan matriks diagonal dengan ukuran  $4 \times 4$ .

ii) tahap prediksi:

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kovarian error} &: P_{k+1}^{f-} = A_k P_k^f A_k^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^{f-} = A_k \hat{x}_k^f + B_k u_k \end{aligned}$$

iii) tahap koreksi:

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kalman gain} &: K_{k+1}^f = P_{k+1}^{f-} H_{k+1}^T (H_{k+1} P_{k+1}^{f-} H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^f = (I - K_{k+1}^f H_{k+1}) P_{k+1}^{f-} \\ \text{estimasi} &: \hat{x}_{k+1}^f = \hat{x}_{k+1}^{f-} + K_{k+1}^f (z_{k+1} - H_{k+1} \hat{x}_{k+1}^{f-}) \end{aligned}$$

- Backward Filter

Pada tahap ini, proses estimasi berjalan mundur dari waktu akhir ke awal.

i) inisialisasi:

Pada langkah ini diberikan inisialisasi awal untuk estimasi waktu akhir ( $\hat{y}_N^{f-}$ ) dan kovarian ( $S_N^{f-}$ ) yaitu

$$\hat{y}_N^- = 0, S_N^- = 0$$

ii) tahap koreksi:

Pada tahap koreksi dihitung kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kovarian error} &: S_{k+1} = S_{k+1}^- + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \\ \text{estimasi} &: \hat{y}_{k+1} = \hat{y}_{k+1}^- + H_{k+1}^T R_{k+1}^{-1} z_{k+1} \end{aligned}$$

iii) tahap prediksi:

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kalman gain} &: K_k^b = S_{k+1} G_k (G_k^T S_{k+1} G_k + Q_k^{-1})^{-1} \\ \text{kovarian error} &: S_k^- = A_k^T (I - K_k^b G_k^T) S_{k+1} A_k \\ \text{estimasi} &: \hat{y}_k^- = A_k^T (I - K_k^b G_k^T) (\hat{y}_{k+1} - S_{k+1} B_k u_k) \end{aligned}$$

- Smoother

Pada tahap smoother dilakukan perhitungan berdasarkan pada tahap Kalman Filter dan information filter.

$$\begin{aligned} \text{kalman gain} &: K_k = P_k^f S_k^- (I + P_k^f S_k^-)^{-1} \\ \text{kovarian error} &: P_k = (I - K_k) P_k^f \\ \text{estimasi} &: \hat{x}_k = (I - K_k) \hat{x}_k^f + P_k \hat{y}_k^- \end{aligned}$$

#### 4.5 Implementasi *Optimal Smoothing* Fuzzy Kalman Filter

Pada *optimal smoothing* Fuzzy Kalman Filter ini digunakan untuk penghalusan hasil estimasi dari metode Fuzzy Kalman Filter. *Optimal smoothing* ini dibutuhkan data pengukuran dari Fuzzy Kalman Filter. *Optimal smoothing* Fuzzy Kalman Filter juga dilakukan proses fuzzifikasi sesuai Fuzzy Kalman Filter yaitu terdapat tiga variabel  $u, w, q$ . Dan juga menentukan aturan dasar yang sama dari sebelumnya dengan 8 aturan. Sehingga proses perhitungan *optimal smoothing* Fuzzy Kalman Filter, yaitu

Rule ke- $i$  :

- model sistem dan model pengukuran

$$\begin{aligned}x_{k+1}^i &= A^i x_k + B u_k + G \xi_k \\z_k &= H x_k + \eta_k\end{aligned}$$

$$x_0 \sim N(\bar{x}_0, P_{x_0}), \xi_k \sim N(0, Q_k), \eta_k \sim N(0, R_k)$$

dimana

$$A^i = \begin{bmatrix} X_u \Delta t + 1 & X_w \Delta t & -\Delta t w_k^i & -g \Delta t \\ (Z_u + q_k^i) \Delta t & Z_w \Delta t + 1 & U_0 \Delta t & 0 \\ \check{M}_u \Delta t & \check{M}_w \Delta t & (\check{M}_q + M_{\dot{w}} u_k) \Delta t + 1 & 0 \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

- Forward Filter

Pada tahap ini, proses estimasi berjalan maju dari awal hingga akhir.

i) inisialisasi:

Pada langkah ini diberikan inisialisasi awal untuk estimasi ( $\hat{x}_0^f$ ) dan kovarian ( $P_0^f$ ) yaitu

$$\hat{x}_0^f = \bar{x}_0, P_0^f = P_0$$

ii) tahap prediksi:

Pada tahap prediksi dihitung kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned}\text{kovarian error} &: P_{k+1}^{i,f-} = A_k^i P_k^{i,f} (A_k^i)^T + G_k Q_k G_k^T \\ \text{estimasi} &: \hat{x}_{k+1}^{i,f-} = A_k^i \hat{x}_k^f + B_k u_k\end{aligned}$$

iii) tahap koreksi:

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned}\text{kalman gain} &: K_{k+1}^{i,f} = P_{k+1}^{i,f-} H_{k+1}^T (H_{k+1} P_{k+1}^{i,f-} H_{k+1}^T + R_{k+1})^{-1} \\ \text{kovarian error} &: P_{k+1}^{i,f} = (I - K_{k+1}^{i,f} H_{k+1}) P_{k+1}^{i,f-} \\ \text{estimasi} &: \hat{x}_{k+1}^{i,f} = \hat{x}_{k+1}^{i,f-} + K_{k+1}^{i,f} (z_{k+1} - H_{k+1} \hat{x}_{k+1}^{i,f-})\end{aligned}$$

- Backward Filter

Pada tahap ini, proses estimasi berjalan mundur dari waktu akhir ke awal.

i) inisialisasi:

Pada langkah ini diberikan inisialisasi awal untuk estimasi waktu akhir ( $\hat{y}_N^{f-}$ ) dan kovarian ( $S_N^{f-}$ ) yaitu

$$\hat{y}_N^- = 0, S_N^- = 0$$

ii) tahap koreksi:

Pada tahap koreksi dihitung kovarian error dan estimasi melalui model pengukuran yaitu

$$\begin{aligned} \text{kovarian error} &: S_{k+1}^i = S_{k+1}^{i,-} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \\ \text{estimasi} &: \hat{y}_{k+1}^i = \hat{y}_{k+1}^{i,-} + H_{k+1}^T R_{k+1}^{-1} z_{k+1} \end{aligned}$$

iii) tahap prediksi:

Pada tahap koreksi dihitung kalman gain, kovarian error dan estimasi melalui model sistem yaitu

$$\begin{aligned} \text{kalman gain} &: K_k^{i,b} = S_{k+1}^i G_k (G_k^T S_{k+1}^i G_k + Q_k^{-1})^{-1} \\ \text{kovarian error} &: S_k^{i,-} = (A_k^i)^T (I - K_k^{i,b} G_k^T) S_{k+1}^i A_k^i \\ \text{estimasi} &: \hat{y}_k^{i,-} = (A_k^i)^T (I - K_k^{i,b} G_k^T) (\hat{y}_{k+1}^i - S_{k+1}^i B_k u_k) \end{aligned}$$

- Smoother

Pada tahap smoother dilakukan perhitungan berdasarkan pada tahap Fuzzy Kalman Filter dan information filter.

$$\begin{aligned} \text{kalman gain} &: K_k^i = P_k^{i,f} S_k^{i,-} (I + P_k^{i,f} S_k^{i,-})^{-1} \\ \text{kovarian error} &: P_k^i = (I - K_k^i) P_k^{i,f} \\ \text{estimasi} &: \hat{x}_k^i = (I - K_k^i) \hat{x}_k^{i,f} + P_k^{i,f} \hat{y}_k^{i,-} \end{aligned}$$

dimana  $i = 1, 2, \dots, 8$ .

Selanjutnya dilakukan proses defuzzifikasi untuk memperoleh hasil estimasi akhir dari kedelapan data tersebut.

## 4.6 Simulasi

Pada simulasi ini dilakukan penerapan metode Kalman Filter dan Fuzzy Kalman Filter. Hasil simulasi ini akan dibandingkan antara metode Kalman Filter dan Fuzzy Kalman Filter terhadap data pengukuran yang bersifat stokastik. Pada simulasi ini juga mempertimbangkan nilai *Root mean square error* (RMSE) antara metode Kalman Filter dan Fuzzy Kalman Filter. Pada penelitian ini, error model pada variabel Fuzzy yang digunakan yaitu 10% dari kondisi awal [4],[11],[14]. Berikut ini adalah anggota dari variabel Fuzzy yaitu:

$$\begin{aligned} u &\in [u - 10\% u, u + 10\% u] \\ w &\in [w - 10\% w, w + 10\% w] \\ q &\in [q - 10\% q, q + 10\% q] \end{aligned}$$

Sehingga diperoleh

$$\begin{aligned} u^- &= 0.9 u \quad , \quad u^+ = 1.1 u \\ w^- &= 0.9 w \quad , \quad w^+ = 1.1 w \\ q^- &= 0.9 q \quad , \quad q^+ = 1.1 q \end{aligned}$$

Pada simulasi ini akan dilakukan beberapa percobaan dengan kasus yang berbeda. Sistem model gerak longitudinal pesawat terbang yang meliputi kecepatan translasi ke depan, kecepatan translasi ke atas, kecepatan sudut *pitch* dan sudut *pitch* yang telah didiskritkan dengan nilai perubahan waktu  $\Delta t = 0.01$  dan nilai untuk kontrol diberikan yaitu  $\delta_e = \delta_T = 0.1$ . Dan juga diberikan nilai kovarian model, kovarian dari noise sistem dan kovarian dari noise pengukuran masing-masing yaitu sebagai berikut:

kovarian model :

$$P_{x_0} = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix}$$

dimana nilai-nilai varians P yang digunakan untuk simulasi yaitu

$$\begin{aligned} P_1 &= 0.5 \\ P_2 &= 0.5 \\ P_3 &= 0.5 \times \pi/180 \\ P_4 &= 0.5 \times \pi/180 \end{aligned}$$

kovarian noise sistem :

$$Q_k = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix}$$

dimana nilai-nilai varians Q yang digunakan untuk simulasi yaitu

$$\begin{aligned} Q_1 &= 0.01 \\ Q_2 &= 0.01 \times \pi/180 \\ Q_3 &= 0.0001 \times \pi/180 \end{aligned}$$

kovarian noise pengukuran :

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

dimana nilai-nilai varians R yang digunakan untuk simulasi yaitu

$$\begin{aligned} R_1 &= 0.01 \\ R_2 &= 0.01 \times \pi/180 \end{aligned}$$

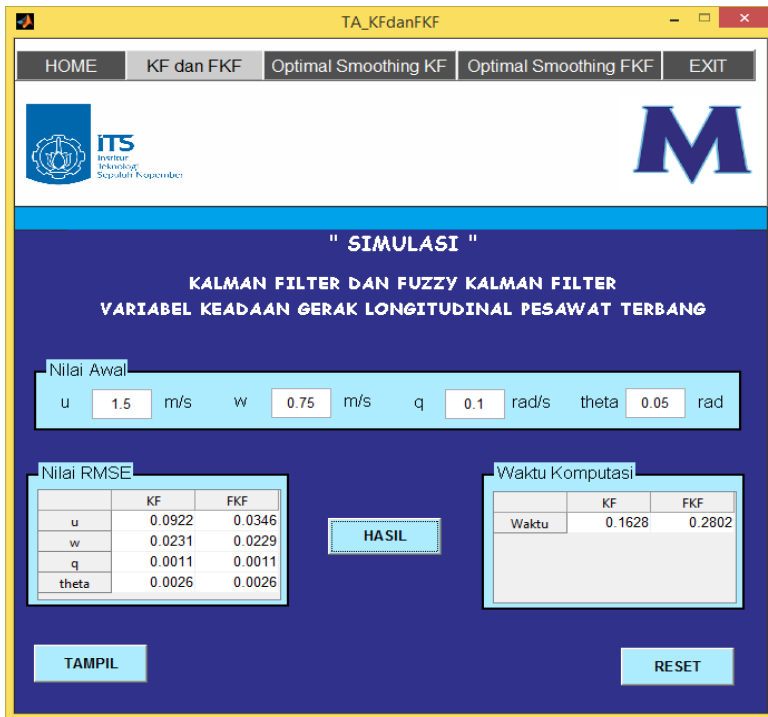
Simulasi estimasi variabel gerak longitudinal pesawat terbang dengan menggunakan program GUI Matlab. Simulasi estimasi variabel gerak pesawat terbang ini ditunjukkan berdasarkan masing-masing variabelnya yang meliputi kecepatan translasi ke depan ( $u$ ), kecepatan translasi ke atas ( $w$ ), kecepatan sudut *pitch* ( $q$ ) dan sudut *pitch* ( $\theta$ ). Berikut ini pada Gambar 4.2 adalah bentuk tampilan utama dari program simulasi estimasi variabel gerak longitudinal pesawat terbang.



Gambar 4.2: Bentuk tampilan GUI awal untuk simulasi estimasi gerak pesawat terbang

#### 4.6.1 Kasus 1

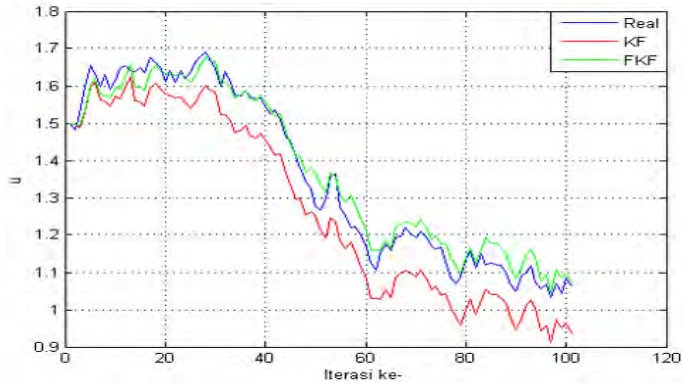
Simulasi pada kasus pertama dengan diberikan kondisi awal yaitu  $u(0) = 1.5 \text{ m/s}$ ,  $w(0) = 0.75 \text{ m/s}$ ,  $q(0) = 0.1 \text{ rad/s}$ ,  $\theta(0) = 0.05 \text{ rad}$ . Dan pada simulasi ini dilakukan *running* sebanyak 10 kali, selanjutnya diambil nilai rata-rata dari sepuluh hasil *running* tersebut. Pada kasus pertama diperoleh hasil estimasi antara metode Kalman Filter dan Fuzzy Kalman Filter masing-masing variabel. Berikut ini ditunjukkan pada Gambar 4.3.



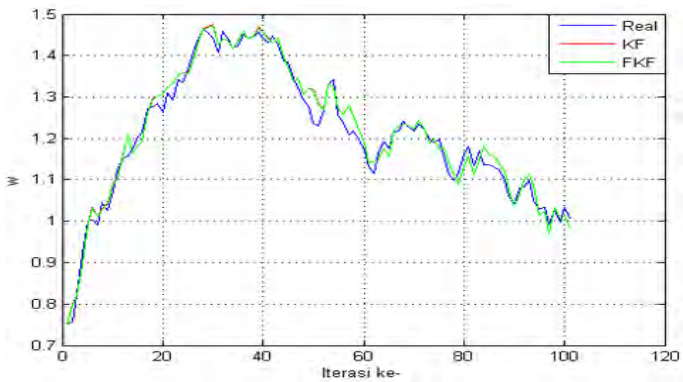
Gambar 4.3: Hasil estimasi variabel gerak longitudinal pesawat terbang



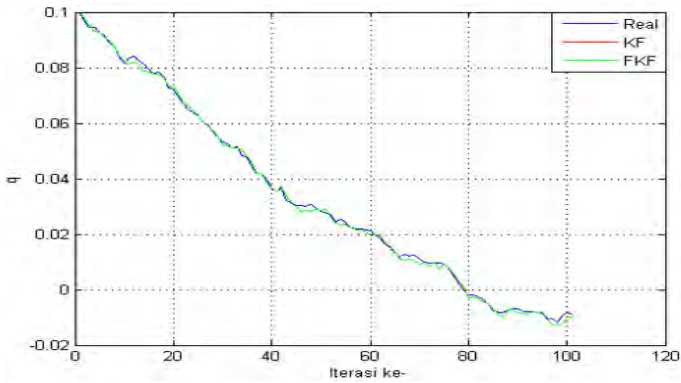
Hasil estimasi variabel keadaan gerak longitudinal pesawat terbang masing-masing, sebagai berikut:



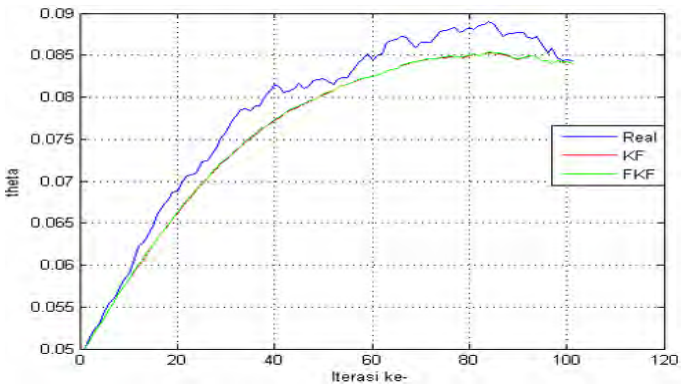
Gambar 4.4: Grafik estimasi kecepatan translasi ke depan ( $u$ )



Gambar 4.5: Grafik estimasi kecepatan translasi ke atas ( $w$ )



Gambar 4.6: Grafik estimasi kecepatan sudut *pitch* ( $q$ )



Gambar 4.7: Grafik estimasi sudut *pitch* ( $\theta$ )

Berdasarkan pada Gambar 4.4 - 4.7, terlihat bahwa metode Fuzzy Kalman Filter memberikan hasil estimasi cenderung relatif lebih akurat dibandingkan dengan metode Kalman Filter biasa. Namun, pada variabel kecepatan translasi ke atas ( $w$ ) dan kecepatan sudut *pitch* ( $q$ ) tidak terlihat jelas. Hal ini dapat diamati dengan data nilai *root mean square error* (RMSE) yang ditunjukkan pada Tabel 4.2. Nilai RMSE menunjukkan bahwa setiap variabel gerak

Tabel 4.2: Nilai RMSE Kalman Filter dan Fuzzy Kalman Filter

Variabel	RMSE		Persentase Peningkatan RMSE
	KF	FKF	
$u$	0.092166	0.034619	62.4 %
$w$	0.023070	0.022908	0.7 %
$q$	0.00109609	0.00109599	0.009 %
$\theta$	0.002600	0.002555	1.7 %

longitudinal pesawat terbang dengan metode Fuzzy Kalman Filter tingkat errornya lebih kecil daripada metode Kalman Filter. Hal ini mengakibatkan tingkat keakurasian dari metode Fuzzy Kalman Filter lebih baik dibandingkan Kalman Filter biasa. Berikut ini adalah tabel perbandingan perbedaan nilai error estimasi masing-masing variabel pada Tabel 4.3.

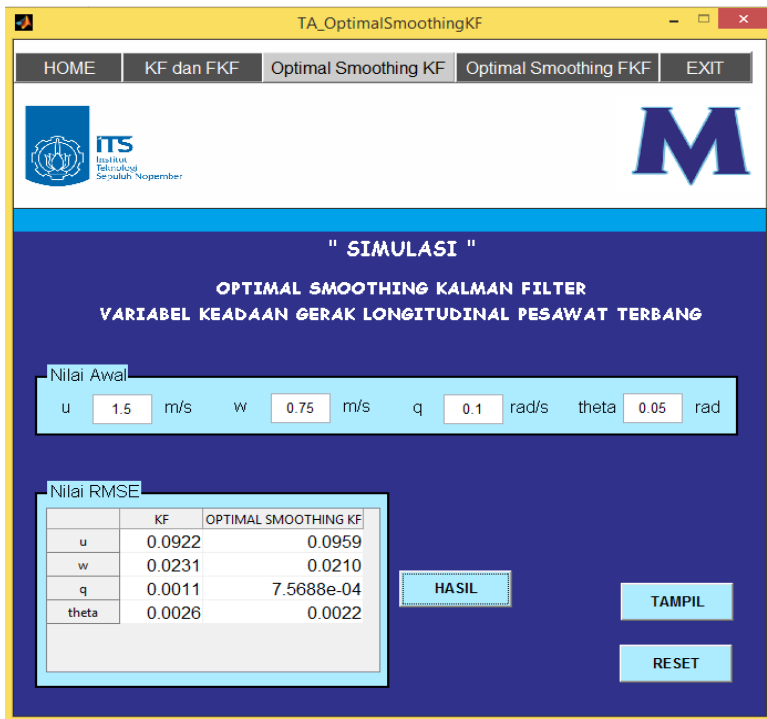
Tabel 4.3: Nilai Error Estimasi Kalman Filter dan Fuzzy Kalman Filter

Variabel	Error Min		Error Max	
	KF	FKF	KF	FKF
$u$	0.012859	0.000178	0.132260	0.091977
$w$	0.000101	0.000485	0.080463	0.080081
$q$	0.000005	0.000005	0.002746	0.002746
$\theta$	0.000089	0.000085	0.004391	0.004303

Selain itu, waktu komputasi yang diperoleh untuk metode Kalman Filter membutuhkan total waktu sebesar 0.1628 s, sedangkan untuk metode Fuzzy Kalman Filter membutuhkan total waktu sebesar 0.2802 s. Berdasarkan waktu komputasi, metode Kalman Filter lebih cepat dibandingkan Fuzzy Kalman Filter.

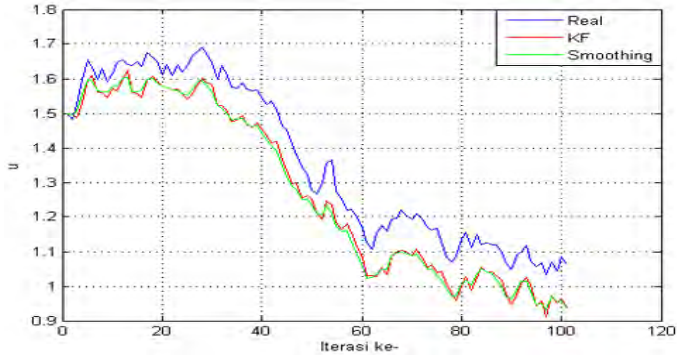
#### 4.6.2 Kasus 2 (*Optimal Smoothing* Kalman Filter)

Pada kasus kedua ini merupakan *optimal smoothing* dari metode Kalman Filter. Berdasarkan kondisi awal pada kasus 1 yaitu  $u(0) = 1.5 \text{ m/s}$ ,  $w(0) = 0.75 \text{ m/s}$ ,  $q(0) = 0.1 \text{ rad/s}$ ,  $\theta(0) = 0.05 \text{ rad}$ . Data pengukuran  $z_k$  diperoleh dari rata-rata kasus pertama. Konsep dari *optimal smoothing* terdiri dari tiga bagian yaitu Kalman Filter, information filter dan smoother. Dengan memilih menu "Optimal Smoothing KF" pada tampilan awal, hasil smoothing Kalman Filter dapat dilihat pada Gambar 4.8.

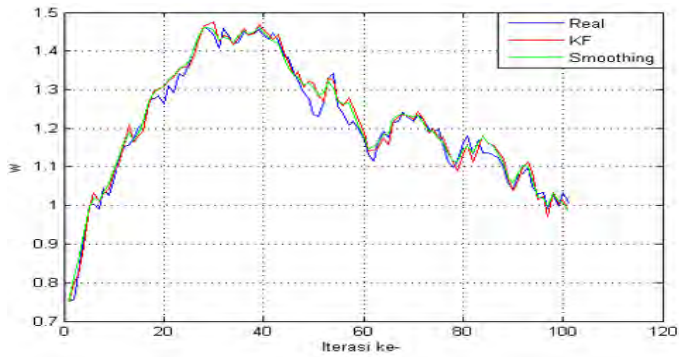


Gambar 4.8: Hasil *optimal smoothing* Kalman Filter untuk variabel gerak longitudinal pesawat terbang

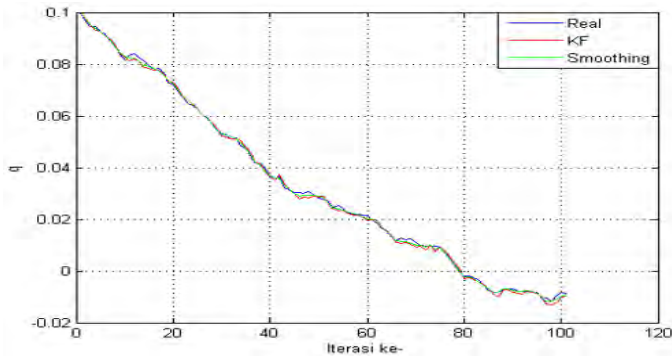
Hasil *optimal smoothing* Kalman Filter pada masing-masing variabel yaitu, sebagai berikut:



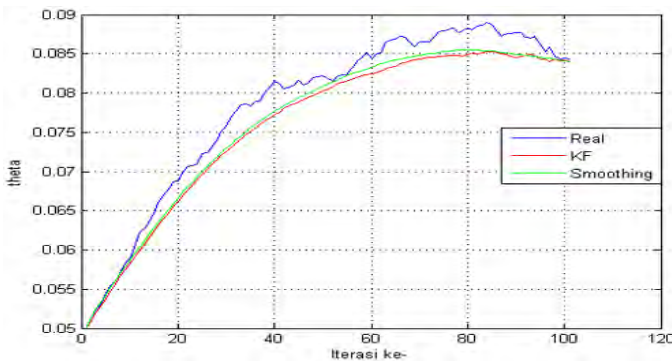
Gambar 4.9: Grafik *optimal smoothing* Kalman Filter pada kecepatan translasi ke depan ( $u$ )



Gambar 4.10: Grafik *optimal smoothing* Kalman Filter pada kecepatan translasi ke atas ( $w$ )



Gambar 4.11: Grafik *optimal smoothing* Kalman Filter pada kecepatan sudut *pitch* ( $q$ )



Gambar 4.12: Grafik *optimal smoothing* Kalman Filter pada sudut *pitch* ( $\theta$ )

Berdasarkan pada Gambar 4.9 - 4.12, terlihat bahwa *optimal smoothing* Kalman Filter menghasilkan hasil yang lebih halus, kecuali pada variabel kecepatan translasi ke depan. Hal ini dapat diperoleh perbedaan nilai *root mean square error* (RMSE) yang ditunjukkan pada Tabel 4.4. Nilai RMSE menunjukkan bahwa *optimal smoothing* pada variabel kecepatan translasi ke atas, kecepatan sudut *pitch* dan sudut

Tabel 4.4: Nilai RMSE Kalman Filter dan Optimal Smoothing

Variabel	RMSE		Persentase Peningkatan RMSE
	KF	Smoothing	
$u$	0.092166	0.095872	-
$w$	0.023070	0.020975	9.1 %
$q$	0.001096	0.000757	30.9 %
$\theta$	0.002600	0.002205	15.2 %

$pitch$  yang lebih kecil nilai errornya. Sedangkan pada variabel kecepatan translasi ke depan, *optimal smoothing* tidak lebih baik dari Kalman Filter biasa. Hal ini dapat dikatakan "*smoothable states*". Sedangkan nilai error minimum dan maksimum estimasi ditunjukkan oleh Tabel 4.5.

Tabel 4.5: Nilai Error Estimasi Kalman Filter dan Optimal Smoothing

Variabel	Error Min		Error Max	
	KF	Smoothing	KF	Smoothing
$u$	0.012859	0.013834	0.132260	0.153197
$w$	0.000101	0.000172	0.080463	0.069039
$q$	0.000005	0.000000	0.002746	0.001926
$\theta$	0.000089	0.000033	0.004391	0.003989

### 4.6.3 Kasus 3 (*Optimal Smoothing Fuzzy Kalman Filter*)

Pada kasus ketiga ini merupakan *optimal smoothing* dari metode Fuzzy Kalman Filter yang juga berdasarkan kondisi awal pada kasus 1. Pada *optimal smoothing* Fuzzy Kalman Filter ini juga memerlukan data pengukuran  $z_k$  yang sudah ada. Proses *optimal smoothing* ini berjalan juga melalui proses

fuzzifikasi, penentuan aturan dasar logika Fuzzy dan tahap *smoothing* dengan penggabungan dari filter berjalan maju dan mundur yang selanjutnya dilakukan defuzzifikasi. Dengan memilih menu "Optimal Smoothing FKF" pada tampilan awal, hasil smoothing Fuzzy Kalman Filter dapat dilihat pada Gambar 4.13.

TA\_OptimalSmoothingFKF

HOME KF dan FKF Optimal Smoothing KF Optimal Smoothing FKF EXIT

ITS Institut Teknologi Sepuluh Nopember

M

" SIMULASI "

OPTIMAL SMOOTHING FUZZY KALMAN FILTER  
VARIABEL KEADAAN GERAK LONGITUDINAL PESAWAT TERBANG

Nilai Awal

u 1.5 m/s w 0.75 m/s q 0.1 rad/s theta 0.05 rad

Nilai RMSE

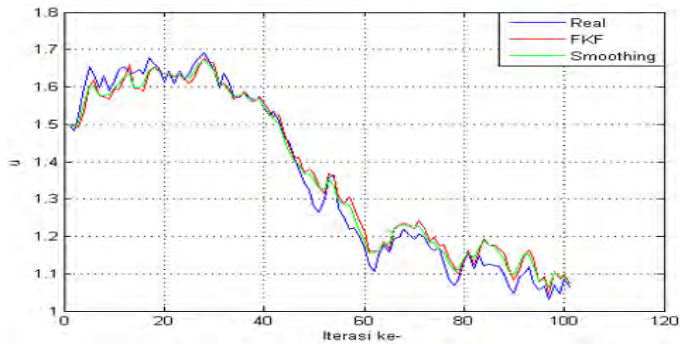
	FKF	OPTIMAL SMOOTHING FKF
u	0.0346	0.0297
w	0.0229	0.0210
q	0.0011	7.5616e-04
theta	0.0026	0.0022

HASIL TAMPIL RESET

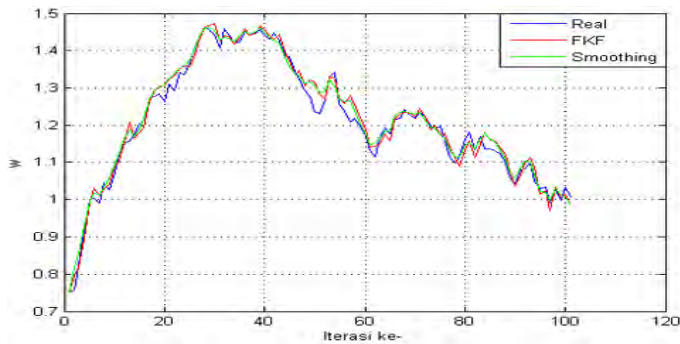
Gambar 4.13: Hasil *optimal smoothing* Fuzzy Kalman Filter untuk variabel gerak longitudinal pesawat terbang



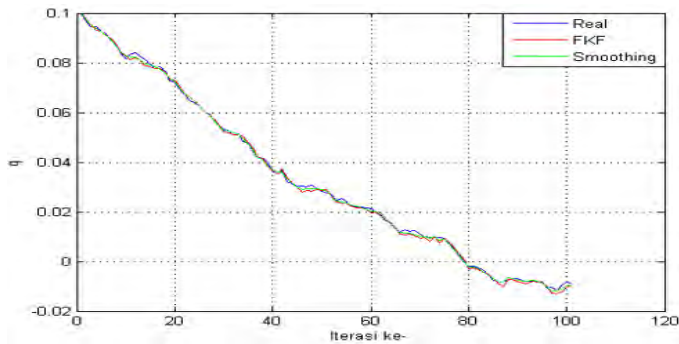
Dan untuk grafik hasil *optimal smoothing* Fuzzy Kalman Filter masing-masing variabel ditunjukkan oleh gambar berikut.



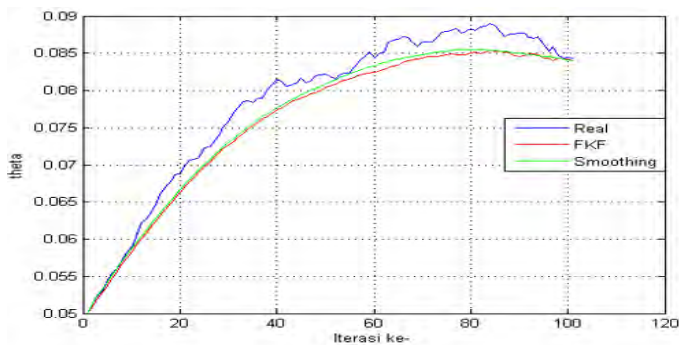
Gambar 4.14: Grafik *optimal smoothing* Fuzzy Kalman Filter pada kecepatan translasi ke depan ( $u$ )



Gambar 4.15: Grafik *optimal smoothing* Fuzzy Kalman Filter pada kecepatan translasi ke atas ( $w$ )



Gambar 4.16: Grafik *optimal smoothing* Fuzzy Kalman Filter pada kecepatan sudut *pitch* ( $q$ )



Gambar 4.17: Grafik *optimal smoothing* Fuzzy Kalman Filter pada sudut *pitch* ( $\theta$ )

Berdasarkan pada Gambar 4.14 - 4.17, terlihat bahwa *optimal smoothing* Fuzzy Kalman Filter menghasilkan hasil yang lebih halus pada semua variabel. Pada kasus ini, *optimal smoothing* berjalan dengan baik. Hal ini dapat dilihat juga perbedaan nilai *root mean square error* (RMSE) yang ditunjukkan pada Tabel 4.6. Nilai RMSE menunjukkan bahwa *optimal smoothing* memberikan nilai errornya lebih kecil pada semua variabel. Sedangkan nilai error minimum

Tabel 4.6: Nilai RMSE Fuzzy Kalman Filter dan Optimal Smoothing

Variabel	RMSE		Persentase Peningkatan RMSE
	FKF	Smoothing	
$u$	0.034619	0.022908	33.8 %
$w$	0.022908	0.021009	8.3 %
$q$	0.001096	0.000756	31 %
$\theta$	0.002555	0.002188	14.4 %

dan maksimum estimasi ditunjukkan oleh Tabel 4.7.

Tabel 4.7: Nilai Error Estimasi Fuzzy Kalman Filter dan Optimal Smoothing

Variabel	Error Min		Error Max	
	FKF	Smoothing	FKF	Smoothing
$u$	0.000178	0.000041	0.091977	0.074399
$w$	0.000485	0.000166	0.080081	0.069026
$q$	0.000005	0.000001	0.002746	0.001918
$\theta$	0.000085	0.000015	0.004303	0.003970

Terlihat bahwa error estimasi maksimum pada *optimal smoothing* Fuzzy Kalman Filter yaitu lebih kecil pada semua variabel dan begitu juga pada error estimasi minimumnya.

*"Halaman ini sengaja dikosongkan."*

## BAB V

### PENUTUP

Pada bab ini, diberikan kesimpulan yang diperoleh dari Tugas Akhir ini serta saran untuk penelitian selanjutnya.

#### 5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat disimpulkan beberapa hal berikut:

1. Hasil estimasi menunjukkan bahwa setiap variabel gerak longitudinal pesawat terbang metode Fuzzy Kalman Filter lebih baik daripada metode Kalman Filter. Nilai RMSE juga menunjukkan bahwa error rata-rata dari metode Fuzzy Kalman Filter lebih kecil daripada metode Kalman Filter. Peningkatan error kecepatan translasi ke depan 62.4 %, kecepatan translasi ke atas 0.7 %, kecepatan sudut *pitch* 0.009 % dan sudut *pitch* 1.7 %.
2. Berdasarkan waktu komputasi menunjukkan bahwa metode Kalman Filter hanya membutuhkan waktu 0.1628s lebih baik daripada metode Fuzzy Kalman Filter yang membutuhkan waktu 0.2802 s.
3. Pada *optimal smoothing* Kalman Filter, hanya variabel kecepatan translasi ke atas, kecepatan sudut *pitch* dan sudut *pitch* yang dapat memberikan hasil estimasi lebih halus daripada Kalman Filter dengan peningkatan error yaitu kecepatan translasi ke atas 9.1 %, kecepatan sudut *pitch* 30.9 % dan sudut *pitch* 15.2 %. Pada variabel kecepatan translasi ke depan, hasil estimasi

Kalman Filter lebih baik dibandingkan pada saat *optimal smoothing*.

4. Pada *optimal smoothing* Fuzzy Kalman Filter, semua variabel dapat memberikan hasil estimasi yang lebih halus daripada Fuzzy Kalman Filter. Peningkatan error masing-masing yaitu kecepatan translasi ke depan 33.8 %, kecepatan translasi ke atas 8.3 %, kecepatan sudut *pitch* 31 % dan sudut *pitch* 14.4 %.

## 5.2 Saran

Pada Tugas Akhir ini, model gerak longitudinal pesawat terbang merupakan sistem model nonlinear. Oleh karena itu, untuk penelitian selanjutnya dapat digunakan dengan metode Extended Kalman Filter. Dengan melakukan kombinasi metode Logika Fuzzy dan Extended Kalman Filter.

## DAFTAR PUSTAKA

- [1] Kallapur, A. G. (2007). "UAV Linear and Nonlinear Estimation Using Eextended Kalman Filter". University of New South Wales.
- [2] Purwanto, E. B. (2003). "Pemodelan Sistem dan Analisis Kestabilan Dinamik Pesawat UAV". Jurnal Teknologi Dirgantara. Laporan Penelitian, LAPAN.
- [3] Salman, S. A. Sreenatha, A. G. Choi, J. T. (2006). "Attitude Dynamics Identification of Unmanned Aircraft Vehicle". Internasional Journal of Control, Automation and System.
- [4] Mahmuri, H. (2011). "Estimasi Perkembangan Sel Kanker Menggunakan Fuzzy Kalman Filter". Jurusan Matematika Institut Teknologi Sepuluh Nopember.
- [5] Sukandi, A. (2010). "Pengendalian Gerak Longitudinal Pesawat Terbang dengan Metode *Decoupling*". Jurusan Teknik Mesin Politeknik Negeri Jakarta.
- [6] Artono, E. (2010). "Pengembangan Sistem Pengendali Posisi Sudut dari Sirip Elevator pada Pesawat Tanpa Awak". Fakultas Teknik Universitas Indonesia.
- [7] Zimmermann, H. J. (1992). **Fuzzy Set Theory and Its Applications**. Second Revised Edition. United States.
- [8] Lewis, F. L. (1998). **Optimal Estimation with An Introduction to Stochastic Control Theory**.

School of Electrical Engineering Georgia Institute of Technology Atlanta. Georgia.

- [9] Han, L. R. (2004). "A Fuzzy-Kalman Filtering Strategy for State Estimation". A Thesis submitted to the College of Graduate Studies and Research in partial Fulfillment of the Requirements for the the Degree of Master of Science. In the Department of Mechanical Engineering University of Saskatchewan. Saskaton, Kanada.
- [10] McLean, D. (1990). **Automatic Flight Control Systems**. Prentice Hall International. UK.
- [11] Chen, G. Xie, Q. Shieh, L. S. (1997). "Fuzzy Kalman Filter". Journal Information of Information Sciences. No. 109, hal. 197-209.
- [12] Purnawan, H. (2015). "Desain Sistem Kendali linear Quadratic Regulator (LQR) untuk Kestabilan Terbang LSU-05". Tugas Akhir. Jurusan Matematika Institut Teknologi Sepuluh Nopember.
- [13] Muzaki, M.A.F. (2015). "Desain Kendali Sudut *Pitch* dan *Roll* pada LSU-05 dengan Metode *Sliding Mode Control (SMC)*". Tugas Akhir. Jurusan Matematika Institut Teknologi Sepuluh Nopember.
- [14] Ermayanti, Z. (2015). "Estimasi dan Kendali Posisi *Autonomous Underwater Vehicle* Berdasarkan Lintasan yang Ditentukan". Jurusan Matematika Institut Teknologi Sepuluh Nopember.



LAMPIRAN

*"Halaman ini sengaja dikosongkan."*

## LAMPIRAN A

### Source Code

#### 1. Program untuk estimasi metode Kalman Filter dan Fuzzy Kalman Filter

i)

```
clear all;
clc;

u0=1.5; w0=0.75; q0=0.1; theta0=0.05;
% parameter-parameter
k = 100; deltat=0.01;
U0=27.78; Xu=0.0002; Xw=0.039; g=9.8; Xde=0.44; Xdt=3.434*10^(-6);
Zu=-0.07; Zw=-0.317; Zde=-5.46; Zdt=-1.5*10^(-7); Mu=8.8*10^(-5);
Mw=-0.002873; Mg=-0.439; Mwdot=-0.0004; Mde=-1.1578; Mdt=6.701*10^(-7);
de=0.1; dt=0.1;
% Model sistem
G=[1 0 0; 1 0 0; 0 0 1];
uk=[de;dt];
Q=[0.01 0 0; 0 0 0.001*pi/180 0; 0 0 0.0001*pi/180];
% Model pengukuran
H=[0 1 0 0; 0 0 1 0];
R=[0.01 0; 0 0.001*pi/180];
% Inisialisasi
x0=[u0;w0;q0;theta0];
u_re=u0; w_re=w0; q_re=q0; theta_re=theta0;
u_kf=u0; w_kf=w0; q_kf=q0; theta_kf=theta0;
u_fkf=u0; w_fkf=w0; q_fkf=q0; theta_fkf=theta0;

x_re0=x0; x_rea=x0; z0=[0;0]; z_re0=[0;0];
xcor_1=x0; xcor_2=x0; xcor_3=x0; xcor_4=x0; xcor_5=x0; xcor_6=x0;
xcor_7=x0; xcor_8=x0; xcor_kf(:,1)=x0; xcor_kf(:,2)=x0;
xcoro_1=x0; xcoro_2=x0; xcoro_3=x0; xcoro_4=x0; xcoro_5=x0;
xcoro_6=x0;
xcoro_7=x0; xcoro_8=x0; xcoro_kf=x0; xcoro_kf=x0;

P0=[0.5 0 0 0; 0.5 0 0 0; 0 0 0.5*pi/180 0; 0 0 0 0.5*pi/180];
Pcor_1=P0; Pcor_2=P0; Pcor_3=P0; Pcor_4=P0; Pcor_5=P0; Pcor_6=P0;
Pcor_7=P0; Pcor_8=P0; Pcor_kf=P0; Pcor_kf=P0;

error_kf1=0; error_kf2=0; error_kf3=0; error_kf4=0;
error_kf5=0; error_kf6=0; error_kf7=0; error_kf8=0;
MSE_kf1=0; MSE_kf2=0; MSE_kf3=0; MSE_kf4=0; MSE_kf5=0;
MSE_kf6=0; MSE_kf7=0; MSE_kf8=0; MSE_kf9=0; MSE_kf10=0;

for j=1:10
    % sistem real
    running=j;
    for i=1:k
        A_re=[1+Xu*deltat Xw*deltat -w_re*deltat -
g*deltat; (Zu+q_re)*...
deltat 1+2w*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mg+... Mwdot*u_re)*deltat 0; 0 0 deltat 1];
        B_re=[Xde*deltat Xdt*deltat; Zde*deltat
Zdt*deltat; Mde*deltat;...
Mdt*deltat; 0 0];
```

ii)

```

x_re=A_re*x_re0+B_re*uk+G*sqrt(Q)*randn(3,1);
s=H*x_re+sqrt(R)*randn(2,1);
x_re0=x_re; x_retot=[x_rea x_re]; x_rea=x_retot;
s_retot=[s0 s]; s0=s_retot;
u_re=x_re(1,:); w_re=x_re(2,:); q_re=x_re(3,:);
theta_re=x_re(4,:);
end
u_re=u0; w_re=w0; q_re=q0; theta_re=theta0;
x_re0=x0; x_retotal=[x_rea x_re0]; x_rea=x_retotal;
s_retotal=[s0 s_re0]; s0=s_retotal;
end

for i=1:k+1
    xre(:,i)=(x_retot(:,i)+x_retot(:,i+k+1)+x_retot(:,i+2*(k+1))+...
    x_retot(:,i+3*(k+1))+x_retot(:,i+4*(k+1))+x_retot(:,i+5*(k+1))+...
    x_retot(:,i+6*(k+1))+x_retot(:,i+7*(k+1))+x_retot(:,i+8*(k+1))+...
    x_retot(:,i+9*(k+1)))/10;
    sre(:,i)=(s_retot(:,i)+s_retot(:,i+k+1)+s_retot(:,i+2*(k+1))+...
    s_retot(:,i+3*(k+1))+s_retot(:,i+4*(k+1))+s_retot(:,i+5*(k+1))+...
    s_retot(:,i+6*(k+1))+s_retot(:,i+7*(k+1))+s_retot(:,i+8*(k+1))+...
    s_retot(:,i+9*(k+1)))/10;
end

for i=1:k
    % Kalman Filter
    tic;
    A_kf=[1Xu*deltat (Xw-q_kf)*deltat -w_kf*deltat -
    g*deltat; (Zu+q_kf)*...
    deltat 1+Zw*deltat (U0+u_kf)*deltat 0; (Mu+Mwdot*q_kf)*deltat
    Mw*...
    deltat 1+(Mq+Mwdot*u_kf)*deltat 0; 0 0 deltat 1];
    B_kf=[Xde*deltat Xdt*deltat; Zde*deltat Zdt*deltat; Mde*deltat
    Mdt*...
    deltat; 0 0];
    % Tahap Prediksi (Time Update)
    Ppre_kf=A_kf*Pcor_kf*transpose(A_kf)+G*Q*transpose(G);
    xpre_kf=A_kf*xcor_kf(:,i)+B_kf*uk;
    % Tahap Koreksi (Measurement Update)
    K_kf=Ppre_kf*transpose(H)*inv(H*Ppre_kf*transpose(H)+R);
    Pcor_kf=(eye(4)-K_kf*H)*Ppre_kf;
    xcor_kf(:,i+1)=xpre_kf+K_kf*(sre(:,i+1)-H*xpre_kf);
    xcortot_kf=[xcoro_kf xcor_kf(:,i+1)]; xcoro_kf=xcortot_kf;
    u_kf=xcor_kf(1,:); w_kf=xcor_kf(2,:); q_kf=xcor_kf(3,:);
    q_kf=xcor_kf(3,:); theta_kf=xcor_kf(4,:);
    waktu_kf(i)=toc;

    % Fussy Kalman Filter

```

iii)

```

tic;
u1=0.9*u_fkf; u2=1.1*u_fkf; w1=0.9*w_fkf; w2=1.1*w_fkf;
q1=0.9*q_fkf; q2=1.1*q_fkf;

miu_u1=(u_fkf-u1)/(u2-u1); miu_u2=(u2-u_fkf)/(u2-u1);
miu_w1=(w_fkf-w1)/(w2-w1); miu_w2=(w2-w_fkf)/(w2-w1);
miu_q1=(q_fkf-q1)/(q2-q1); miu_q2=(q2-q_fkf)/(q2-q1);

u_min=u1*miu_u1; u_max=u2*miu_u2; w_min=w1*miu_w1;
w_max=w2*miu_w2;
q_min=q1*miu_q1; q_max=q2*miu_q2;

B_fkf=[Xde*deltat Xdt*deltat; Zde*deltat Zdt*deltat; Mde*deltat...
Mdt*deltat; 0 0];
A_1=[1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat; (Zu+q_min)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_min)*...
deltat 0; 0 0 deltat 1];
A_2=[1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat; (Zu+q_max)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_min)*...
deltat 0; 0 0 deltat 1];
A_3=[1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat; (Zu+q_min)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_min)*...
deltat 0; 0 0 deltat 1];
A_4=[1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat; (Zu+q_max)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_min)*...
deltat 0; 0 0 deltat 1];
A_5=[1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat; (Zu+q_min)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_max)*...
deltat 0; 0 0 deltat 1];
A_6=[1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat; (Zu+q_max)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_max)*...
deltat 0; 0 0 deltat 1];
A_7=[1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat; (Zu+q_min)*deltat...
1+Zw*deltat U0*deltat 0; Mu*deltat Mw*deltat
1+(Mq+Mvdot*u_max)*...
deltat 0; 0 0 deltat 1];
A_8=[1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat; (Zu+q_max)*deltat...

```

iv)

```

1+Zw*deltat UQ*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mw*dot*u_max)*...
deltat 0;0 0 deltat 1];
% Tahap Prediksi (Time Update)
Ppre_1=(A_1*Pcor_1*transpose(A_1))+(G*Q*transpose(G));
Ppre_2=(A_2*Pcor_2*transpose(A_2))+(G*Q*transpose(G));
Ppre_3=(A_3*Pcor_3*transpose(A_3))+(G*Q*transpose(G));
Ppre_4=(A_4*Pcor_4*transpose(A_4))+(G*Q*transpose(G));
Ppre_5=(A_5*Pcor_5*transpose(A_5))+(G*Q*transpose(G));
Ppre_6=(A_6*Pcor_6*transpose(A_6))+(G*Q*transpose(G));
Ppre_7=(A_7*Pcor_7*transpose(A_7))+(G*Q*transpose(G));
Ppre_8=(A_8*Pcor_8*transpose(A_8))+(G*Q*transpose(G));

xpre_1=A_1*xcor_1+B_fkf*uk; xpre_2=A_2*xcor_2+B_fkf*uk;
xpre_3=A_3*xcor_3+B_fkf*uk; xpre_4=A_4*xcor_4+B_fkf*uk;
xpre_5=A_5*xcor_5+B_fkf*uk; xpre_6=A_6*xcor_6+B_fkf*uk;
xpre_7=A_7*xcor_7+B_fkf*uk; xpre_8=A_8*xcor_8+B_fkf*uk;
% Tahap Koreksi (Measurement Update)
K_1=Ppre_1*transpose(H)*inv(H*Ppre_1*transpose(H)+R);
K_2=Ppre_2*transpose(H)*inv(H*Ppre_2*transpose(H)+R);
K_3=Ppre_3*transpose(H)*inv(H*Ppre_3*transpose(H)+R);
K_4=Ppre_4*transpose(H)*inv(H*Ppre_4*transpose(H)+R);
K_5=Ppre_5*transpose(H)*inv(H*Ppre_5*transpose(H)+R);
K_6=Ppre_6*transpose(H)*inv(H*Ppre_6*transpose(H)+R);
K_7=Ppre_7*transpose(H)*inv(H*Ppre_7*transpose(H)+R);
K_8=Ppre_8*transpose(H)*inv(H*Ppre_8*transpose(H)+R);

Pcor_1=(eye(4)-K_1*H)*Ppre_1; Pcor_2=(eye(4)-K_2*H)*Ppre_2;
Pcor_3=(eye(4)-K_3*H)*Ppre_3; Pcor_4=(eye(4)-K_4*H)*Ppre_4;
Pcor_5=(eye(4)-K_5*H)*Ppre_5; Pcor_6=(eye(4)-K_6*H)*Ppre_6;
Pcor_7=(eye(4)-K_7*H)*Ppre_7; Pcor_8=(eye(4)-K_8*H)*Ppre_8;

xcor_1=xpre_1+K_1*(zre(:,i+1)-H*xpre_1);
xcor_2=xpre_2+K_2*(zre(:,i+1)-H*xpre_2);
xcor_3=xpre_3+K_3*(zre(:,i+1)-H*xpre_3);
xcor_4=xpre_4+K_4*(zre(:,i+1)-H*xpre_4);
xcor_5=xpre_5+K_5*(zre(:,i+1)-H*xpre_5);
xcor_6=xpre_6+K_6*(zre(:,i+1)-H*xpre_6);
xcor_7=xpre_7+K_7*(zre(:,i+1)-H*xpre_7);
xcor_8=xpre_8+K_8*(zre(:,i+1)-H*xpre_8);

xcortot_1=[xcoro_1 xcor_1]; xcortot_2=[xcoro_2 xcor_2];
xcortot_3=[xcoro_3 xcor_3]; xcortot_4=[xcoro_4 xcor_4];
xcortot_5=[xcoro_5 xcor_5]; xcortot_6=[xcoro_6 xcor_6];
xcortot_7=[xcoro_7 xcor_7]; xcortot_8=[xcoro_8 xcor_8];
xcoro_1=xcortot_1; xcoro_2=xcortot_2; xcoro_3=xcortot_3;
xcoro_4=xcortot_4; xcoro_5=xcortot_5; xcoro_6=xcortot_6;
xcoro_7=xcortot_7; xcoro_8=xcortot_8;
% Proses Defuzzifikasi
W_1=miu_u1*miu_w1*miu_q1; W_2=miu_u1*miu_w1*miu_q2;
W_3=miu_u1*miu_w2*miu_q1; W_4=miu_u1*miu_w2*miu_q2;

```

v)

```

W_5=miu_u2*miu_w1*miu_q1; W_6=miu_u2*miu_w1*miu_q2;
W_7=miu_u2*miu_w2*miu_q1; W_8=miu_u2*miu_w2*miu_q2;

xcor_fkf(:, :, i+1)=(W_1*xcor_1+W_2*xcor_2+W_3*xcor_3+W_4*xcor_4+...
W_5*xcor_5+W_6*xcor_6+W_7*xcor_7+W_8*xcor_8)/(W_1+W_2+W_3+...
W_4+W_5+W_6+W_7+W_8);
xcortot_fkf=(xcoro_fkf xcor_fkf(:, :, i+1));
xcoro_fkf=xcortot_fkf;
u_fkf=xcor_fkf(1, :, i+1); w_fkf=xcor_fkf(2, :, i+1);
q_fkf=xcor_fkf(3, :, i+1); theta_fkf=xcor_fkf(4, :, i+1);
waktu_fkf(i)=toc;

% Error kecepatan translasi ke depan
error_kf1=[error_kf1 abs(xre(1, :, i+1)-xcor_kf(1, :, i+1))];
error_fkf1=[error_fkf1 abs(xre(1, :, i+1)-xcor_fkf(1, :, i+1))];
% Error kecepatan translasi ke atas
error_kf2=[error_kf2 abs(xre(2, :, i+1)-xcor_kf(2, :, i+1))];
error_fkf2=[error_fkf2 abs(xre(2, :, i+1)-xcor_fkf(2, :, i+1))];
% Error kecepatan sudut pitch
error_kf3=[error_kf3 abs(xre(3, :, i+1)-xcor_kf(3, :, i+1))];
error_fkf3=[error_fkf3 abs(xre(3, :, i+1)-xcor_fkf(3, :, i+1))];
% Error sudut pitch
error_kf4=[error_kf4 abs(xre(4, :, i+1)-xcor_kf(4, :, i+1))];
error_fkf4=[error_fkf4 abs(xre(4, :, i+1)-xcor_fkf(4, :, i+1))];
% MSE kecepatan translasi ke depan
MSE_kf1=MSE_kf1+(xre(1, :, i+1)-xcor_kf(1, :, i+1))^2;
MSE_fkf1=MSE_fkf1+(xre(1, :, i+1)-xcor_fkf(1, :, i+1))^2;
% MSE kecepatan translasi ke atas
MSE_kf2=MSE_kf2+(xre(2, :, i+1)-xcor_kf(2, :, i+1))^2;
MSE_fkf2=MSE_fkf2+(xre(2, :, i+1)-xcor_fkf(2, :, i+1))^2;
% MSE kecepatan sudut pitch
MSE_kf3=MSE_kf3+(xre(3, :, i+1)-xcor_kf(3, :, i+1))^2;
MSE_fkf3=MSE_fkf3+(xre(3, :, i+1)-xcor_fkf(3, :, i+1))^2;
% MSE sudut pitch
MSE_kf4=MSE_kf4+(xre(4, :, i+1)-xcor_kf(4, :, i+1))^2;
MSE_fkf4=MSE_fkf4+(xre(4, :, i+1)-xcor_fkf(4, :, i+1))^2;

end
disp('Error KF kecepatan translasi ke depan');
sprintf('%0.6f' , error_kf1)
disp('Error FKF kecepatan translasi ke depan');
sprintf('%0.6f' , error_fkf1)
disp('Error KF kecepatan translasi ke atas');
sprintf('%0.6f' , error_kf2)
disp('Error FKF kecepatan translasi ke atas');
sprintf('%0.6f' , error_fkf2)
disp('Error KF kecepatan sudut pitch');
sprintf('%0.6f' , error_kf3)
disp('Error FKF kecepatan sudut pitch');
sprintf('%0.6f' , error_fkf3)
disp('Error KF sudut pitch');

```

vi)

```

sprintf('%0.6f' , error_kf4)
disp('Error FKF sudut pitch');
sprintf('%0.6f' , error_fkf4)
disp('RMSE KF kecepatan translasi ke depan');
RMSE_kf1=sqrt(MSE_kf1/k); sprintf('%0.6f',RMSE_kf1)
disp('RMSE FKF kecepatan translasi ke depan');
RMSE_fkf1=sqrt(MSE_fkf1/k); sprintf('%0.6f',RMSE_fkf1)
disp('RMSE KF kecepatan translasi ke atas');
RMSE_kf2=sqrt(MSE_kf2/k); sprintf('%0.6f',RMSE_kf2)
disp('RMSE FKF kecepatan translasi ke atas');
RMSE_fkf2=sqrt(MSE_fkf2/k); sprintf('%0.6f',RMSE_fkf2)
disp('RMSE KF kecepatan sudut pitch');
RMSE_kf3=sqrt(MSE_kf3/k); sprintf('%0.8f',RMSE_kf3)
disp('RMSE FKF kecepatan sudut pitch');
RMSE_fkf3=sqrt(MSE_fkf3/k); sprintf('%0.8f',RMSE_fkf3)
disp('RMSE KF sudut pitch');
RMSE_kf4=sqrt(MSE_kf4/k); sprintf('%0.6f',RMSE_kf4)
disp('RMSE FKF sudut pitch');
RMSE_fkf4=sqrt(MSE_fkf4/k); sprintf('%0.6f',RMSE_fkf4)
disp('Running Time')
time_kf=sum(waktu_kf); time_fkf=sum(waktu_fkf);

figure(1)
plot(1:k+1,xre(1,:), 'blue',1:k+1,xcortot_kf(1,:), 'red',1:k+1,...
     xcortot_fkf(1,:), 'green')
title('Estimasi Kecepatan translasi ke depan')
xlabel('Iterasi ke-')
ylabel('Kecepatan translasi ke depan')
grid on;
legend('Real', 'Estimasi Kalman Filter', 'Estimasi Fussy Kalman Filter');
figure(2)
plot(1:k+1,xre(2,:), 'blue',1:k+1,xcortot_kf(2,:), 'red',1:k+1,...
     xcortot_fkf(2,:), 'green')
title('Estimasi Kecepatan translasi ke atas')
xlabel('Iterasi ke-')
ylabel('Kecepatan translasi ke atas')
grid on;
legend('Real', 'Estimasi Kalman Filter', 'Estimasi Fussy Kalman Filter');
figure(3)
plot(1:k+1,xre(3,:), 'blue',1:k+1,xcortot_kf(3,:), 'red',1:k+1,...
     xcortot_fkf(3,:), 'green')
title('Estimasi Kecepatan sudut pitch')
xlabel('Iterasi ke-')
ylabel('Kecepatan sudut pitch')
grid on;
legend('Real', 'Estimasi Kalman Filter', 'Estimasi Fussy Kalman Filter');
figure(4)
plot(1:k+1,xre(4,:), 'blue',1:k+1,xcortot_kf(4,:), 'red',1:k+1,...

```



vii)

```
xcortot_fkf(4,:),'green')
title('Estimasi Sudut pitch')
xlabel('Iterasi ke-')
ylabel('Sudut pitch')
grid on;
legend('Real','Estimasi Kalman Filter','Estimasi Fuzzy Kalman
Filter');|
```

2. Program untuk estimasi *optimal smoothing* Kalman Filter

i)

```

clear all;
clc;

u0=1.5; w0=0.75; q0=0.1; theta0=0.05;
% parameter-parameter
k = 100; deltat=0.01;
U0=27.78; Xu=0.0002; Xw=0.039; g=3.8; Xda=0.44; Xdt=3.434*10^(-6);
zu=0.07; zw=0.317; Zda=-3.46; Zdt=-1.5*10^(-7); Mu=8.8*10^(-5);
Mw=0.002873; Mg=-0.439; Mwdot=-0.0004; Mda=-1.1578; Mdt=6.701*10^(-7);
da=0.1; dt=0.1;
% Model sistem
Q=[0 0 0; 0 0 0; 0 0 0]; uk=[da;dt];
Q=[0.01 0 0; 0 0.01*pi/180 0; 0 0 0.0001*pi/180];
% Model pengukuran
H=[0 1 0 0; 0 0 1 0]; R=[0.01 0; 0 0.01*pi/180];
% Inisialisasi
x0=[u0;w0;q0;theta0];
u0_kf=u0; w0_kf=w0; q0_kf=q0; theta_kf=theta0;
xfcor_kf(:,1)=x0; xcor_kf(:,1)=x0; xbpres_kf(:,1)=0; x0_kf=0;
P0=[0.5 0 0; 0 0.5 0; 0 0 0.5*pi/180]; P0_kf=P0; P0_kf(:,1)=0;
P0_kf(:,1)=P0; P0_kf(:,1)=P0; P0_kf(:,1)=P0;

error_kf1=0; error_0skf1=0; error_kf2=0; error_0skf2=0;
error_kf3=0; error_0skf3=0; error_kf4=0; error_0skf4=0;
MSE_kf1=0; MSE_0skf1=0; MSE_kf2=0; MSE_0skf2=0; MSE_kf3=0; MSE_0skf3=0;
MSE_kf4=0; MSE_0skf4=0;

% sistem real
xre(:,1)=x0; xre(:,2)=[1.4820;0.7573;0.0964;0.0514]; xre(:,3)=[1.5243;0.82
33;0.0947;0.0523];
xre(:,4)=[1.5979;0.9213;0.0946;0.0530]; xre(:,5)=[1.6544;1.0014;0.0925;0.05
45]; xre(:,6)=[1.6328;1.0027;0.0916;0.0554];
xre(:,7)=[1.5986;0.9911;0.0891;0.0560]; xre(:,8)=[1.6294;1.0439;0.0878;0.05
70]; xre(:,9)=[1.5905;1.0266;0.0838;0.0582];
xre(:,10)=[1.6170;1.0735;0.0817;0.0589]; xre(:,11)=[1.6466;1.1230;0.0835;0.
0602]; xre(:,12)=[1.6540;1.1507;0.0841;0.0622];
xre(:,13)=[1.6403;1.1579;0.0822;0.0627]; xre(:,14)=[1.6372;1.1747;0.0813;0.
0635]; xre(:,15)=[1.6476;1.2049;0.0795;0.0647];
xre(:,16)=[1.6362;1.2129;0.0776;0.0660]; xre(:,17)=[1.6761;1.2716;0.0783;0.
0670]; xre(:,18)=[1.6611;1.2737;0.0765;0.0677];
xre(:,19)=[1.6507;1.2838;0.0728;0.0686]; xre(:,20)=[1.6109;1.2615;0.0721;0.
0688]; xre(:,21)=[1.6415;1.3094;0.0697;0.0698];
xre(:,22)=[1.6092;1.2936;0.0672;0.0707]; xre(:,23)=[1.6416;1.3418;0.0647;0.
0708]; xre(:,24)=[1.6202;1.3355;0.0640;0.0710];
xre(:,25)=[1.6349;1.3651;0.0625;0.0723]; xre(:,26)=[1.6664;1.4110;0.0603;0.
0724]; xre(:,27)=[1.6780;1.4363;0.0596;0.0731];
xre(:,28)=[1.6913;1.4631;0.0569;0.0739]; xre(:,29)=[1.6718;1.4561;0.0553;0.
0751]; xre(:,30)=[1.6455;1.4421;0.0532;0.0758];
xre(:,31)=[1.5984;1.4067;0.0525;0.0771]; xre(:,32)=[1.6384;1.4584;0.0510;0.
0777]; xre(:,33)=[1.6087;1.4398;0.0513;0.0785];
xre(:,34)=[1.5746;1.4172;0.0484;0.0786]; xre(:,35)=[1.5718;1.4250;0.0474;0.
0784]; xre(:,36)=[1.5890;1.4524;0.0439;0.0790];
xre(:,37)=[1.5708;1.4432;0.0420;0.0790]; xre(:,38)=[1.5639;1.4449;0.0416;0.
0801]; xre(:,39)=[1.5678;1.4575;0.0395;0.0808];
xre(:,40)=[1.5458;1.4434;0.0363;0.0815]; xre(:,41)=[1.5256;1.4302;0.0353;0.
0812]; xre(:,42)=[1.5349;1.4463;0.0365;0.0806];

```



iii)

```

x=0(:,:22)-[1.3624;0.0664];x=0(:,:23)-[1.3558;0.0649];x=0(:,:24)-[1.3510;0.
0630];
x=0(:,:25)-[1.4043;0.0592];x=0(:,:26)-[1.4497;0.0596];x=0(:,:27)-[1.4829;0.
0569];
x=0(:,:28)-[1.4657;0.0553];x=0(:,:29)-[1.4746;0.0511];x=0(:,:30)-[1.3951;0.
0526];
x=0(:,:31)-[1.4456;0.0514];x=0(:,:32)-[1.4312;0.0522];x=0(:,:33)-[1.4027;0.
0505];
x=0(:,:34)-[1.4432;0.0471];x=0(:,:35)-[1.4713;0.0450];x=0(:,:36)-[1.4291;0.
0405];
x=0(:,:37)-[1.4460;0.0423];x=0(:,:38)-[1.4809;0.0368];x=0(:,:39)-[1.4509;0.
0368];
x=0(:,:40)-[1.4319;0.0351];x=0(:,:41)-[1.4249;0.0393];x=0(:,:42)-[1.4497;0.
0318];
x=0(:,:43)-[1.3673;0.0309];x=0(:,:44)-[1.3578;0.0303];x=0(:,:45)-[1.3167;0.
0271];
x=0(:,:46)-[1.3531;0.0302];x=0(:,:47)-[1.2840;0.0290];x=0(:,:48)-[1.3312;0.
0300];
x=0(:,:49)-[1.3167;0.0295];x=0(:,:50)-[1.2619;0.0297];x=0(:,:51)-[1.2621;0.
0254];
x=0(:,:52)-[1.3693;0.0234];x=0(:,:53)-[1.3204;0.0240];x=0(:,:54)-[1.2430;0.
0247];
x=0(:,:55)-[1.2508;0.0221];x=0(:,:56)-[1.2955;0.0222];x=0(:,:57)-[1.2444;0.
0225];
x=0(:,:58)-[1.2006;0.0210];x=0(:,:59)-[1.1743;0.0197];x=0(:,:60)-[1.1119;0.
0209];
x=0(:,:61)-[1.1488;0.0201];x=0(:,:62)-[1.1514;0.0165];x=0(:,:63)-[1.1987;0.
0153];
x=0(:,:64)-[1.1476;0.0130];x=0(:,:65)-[1.2558;0.0102];x=0(:,:66)-[1.2425;0.
0112];
x=0(:,:67)-[1.2434;0.0118];x=0(:,:68)-[1.2327;0.0107];x=0(:,:69)-[1.2208;0.
0097];
x=0(:,:70)-[1.2606;0.0104];x=0(:,:71)-[1.2137;0.0080];x=0(:,:72)-[1.1705;0.
0119];
x=0(:,:73)-[1.2103;0.0070];x=0(:,:74)-[1.1619;0.0111];x=0(:,:75)-[1.1873;0.
0080];
x=0(:,:76)-[1.1200;0.0050];x=0(:,:77)-[1.0976;0.0022];x=0(:,:78)-[1.0826;0.
0007];
x=0(:,:79)-[1.1672;-0.0050];x=0(:,:80)-[1.1787;-
0.0018];x=0(:,:81)-[1.0324;-0.0027];
x=0(:,:82)-[1.1804;-0.0044];x=0(:,:83)-[1.2058;-
0.0049];x=0(:,:84)-[1.1352;-0.0071];
x=0(:,:85)-[1.1605;-0.0097];x=0(:,:86)-[1.1402;-
0.0098];x=0(:,:87)-[1.1162;-0.0048];
x=0(:,:88)-[1.0525;-0.0062];x=0(:,:89)-[1.0214;-
0.0081];x=0(:,:90)-[1.0312;-0.0083];
x=0(:,:91)-[1.1275;-0.0083];x=0(:,:92)-[1.1302;-
0.0067];x=0(:,:93)-[1.0780;-0.0076];
x=0(:,:94)-[0.9770;-0.0078];x=0(:,:95)-[1.0383;-
0.0104];x=0(:,:96)-[0.9448;-0.0135];
x=0(:,:97)-[1.0813;-0.0126];x=0(:,:98)-[0.9961;-
0.0109];x=0(:,:99)-[1.0282;-0.0083];
x=0(:,:100)-[0.9719;-0.0083];

```

```

for i=1:k
    % Optimal Smoothing Kalman Filter

```

iv)

```

    A_kf(:,:,1)=[I-Xu*deltat (Xu-q_kf)*deltat -w_kf*deltat -
    q*deltat; (Xu-q_kf)*deltat I+Zu*deltat (UO+u_kf)*deltat
    0; (Mu-Mwdot*u_kf)*deltat Mw*deltat I+(Mq-Mwdot*u_kf)*deltat 0; 0 0 0 deltat I];
    B_kf=[Xde*deltat Xdt*deltat; Zde*deltat Zdt*deltat; Mde*deltat Mdt*deltat; 0
    0];
    % Forward filter
    % Time Update

    Ffpr_kf(:,:,1+1)=A_kf(:,:,1)*Ffcor_kf(:,:,1)*transpose(A_kf(:,:,1))+G*Q*transp
    ose(G);
    xfp_kf(:,:,1+1)=A_kf(:,:,1)*xfcor_kf(:,:,1)+B_kf*uk;
    % Measurement Update

    Kf_kf(:,:,1+1)=Ffpr_kf(:,:,1+1)*transpose(H)*inv(H*Ffpr_kf(:,:,1+1)*transp
    ose(H)+R);
    Ffcor_kf(:,:,1+1)=(eye(4)-Kf_kf(:,:,1+1)*H)*Ffpr_kf(:,:,1+1);
    xfcor_kf(:,:,1+1)=xfpr_kf(:,:,1+1)+Kf_kf(:,:,1+1)*(zme(:,:,1)-
    H*xfpr_kf(:,:,1+1));

    u_kf=xfcor_kf(1,:,1+1); w_kf=xfcor_kf(2,:,1+1);
    q_kf=xfcor_kf(3,:,1+1); theta_kf=xfcor_kf(4,:,1+1);

end
for i=1:k
    % Backward filter
    % Measurement Update
    Fbcor_kf(:,:,k-i+2)=Fbpr_kf(:,:,k-i+2)+transpose(H)*inv(R)*B;
    xbcor_kf(:,:,k-i+2)=xbpr_kf(:,:,k-i+2)+transpose(H)*inv(R)*zme(:,:,k-
    i+1);
    % Time Update
    Kb_kf(:,:,k-i+1)=Fbcor_kf(:,:,k-
    i+2)*G*inv(transpose(G)*Fbcor_kf(:,:,k-i+2)*G+inv(Q));
    Fbpr_kf(:,:,k-i+1)=transpose(A_kf(:,:,k-i+1))* (eye(4)-Kb_kf(:,:,k-
    i+1)*transpose(G))*Fbcor_kf(:,:,k-i+2)*A_kf(:,:,k-i+1);
    xbp_kf(:,:,k-i+1)=transpose(A_kf(:,:,k-i+1))* (eye(4)-Kb_kf(:,:,k-
    i+1)*transpose(G))* (xbcor_kf(:,:,k-i+2)-Fbcor_kf(:,:,k-i+2)*B_kf*uk);
end
for i=1:k
    % Smoother

    K_kf(:,:,i+1)=Ffcor_kf(:,:,i+1)*Fbpr_kf(:,:,i+1)*inv(eye(4)+Ffcor_kf(:,:,i+1)
    *Fbpr_kf(:,:,i+1));
    F_kf(:,:,i+1)=(eye(4)-K_kf(:,:,i+1))*Ffcor_kf(:,:,i+1);
    xcor_kf(:,:,i+1)=(eye(4)-
    K_kf(:,:,i+1))*xfcor_kf(:,:,i+1)+F_kf(:,:,i+1)*xbpr_kf(:,:,i+1);
end
for i=1:k
    % Error kecepatan translasi ke depan
    error_kf1=[error_kf1 abs(xme(1,:,i+1)-xfcor_kf(1,:,i+1))];
    error_OSKf1=[error_OSKf1 abs(xme(1,:,i+1)-xcor_kf(1,:,i+1))];
    % Error kecepatan translasi ke atas
    error_kf2=[error_kf2 abs(xme(2,:,i+1)-xfcor_kf(2,:,i+1))];
    error_OSKf2=[error_OSKf2 abs(xme(2,:,i+1)-xcor_kf(2,:,i+1))];
    % Error kecepatan sudut pitch
    error_kf3=[error_kf3 abs(xme(3,:,i+1)-xfcor_kf(3,:,i+1))];
    error_OSKf3=[error_OSKf3 abs(xme(3,:,i+1)-xcor_kf(3,:,i+1))];
    % Error sudut pitch

```

v)

```

error_kf4=[error_kf4 abs(wra(4,1,i+1)-wfoor_kf(4,1,i+1))];
error_Oskf4=[error_Oskf4 abs(wra(4,1,i+1)-wcoor_kf(4,1,i+1))];
% MSE kecepatan translasi ke depan
MSE_kf1-MSE_kf1+(wra(1,1,i+1)-wfoor_kf(1,1,i+1))^2;
MSE_Oskf1-MSE_Oskf1+(wra(1,1,i+1)-wcoor_kf(1,1,i+1))^2;
% MSE kecepatan translasi ke atas
MSE_kf2-MSE_kf2+(wra(2,1,i+1)-wfoor_kf(2,1,i+1))^2;
MSE_Oskf2-MSE_Oskf2+(wra(2,1,i+1)-wcoor_kf(2,1,i+1))^2;
% MSE kecepatan sudut pitch
MSE_kf3-MSE_kf3+(wra(3,1,i+1)-wfoor_kf(3,1,i+1))^2;
MSE_Oskf3-MSE_Oskf3+(wra(3,1,i+1)-wcoor_kf(3,1,i+1))^2;
% MSE sudut pitch
MSE_kf4-MSE_kf4+(wra(4,1,i+1)-wfoor_kf(4,1,i+1))^2;
MSE_Oskf4-MSE_Oskf4+(wra(4,1,i+1)-wcoor_kf(4,1,i+1))^2;
end
disp('Error KF kecepatan translasi ke depan');
sprintf('%0.6f', error_kf1)
disp('Error OSKF kecepatan translasi ke depan');
sprintf('%0.6f', error_Oskf1)
disp('Error KF kecepatan translasi ke atas');
sprintf('%0.6f', error_kf2)
disp('Error OSKF kecepatan translasi ke atas');
sprintf('%0.6f', error_Oskf2)
disp('Error KF kecepatan sudut pitch');
sprintf('%0.6f', error_kf3)
disp('Error OSKF kecepatan sudut pitch');
sprintf('%0.6f', error_Oskf3)
disp('Error KF sudut pitch');
sprintf('%0.6f', error_kf4)
disp('Error OSKF sudut pitch');
sprintf('%0.6f', error_Oskf4)
disp('RMSE KF kecepatan translasi ke depan');
RMSE_kf1=sgt(MSE_kf1/k); sprintf('%0.6f',RMSE_kf1)
disp('RMSE OSKF kecepatan translasi ke depan');
RMSE_Oskf1=sgt(MSE_Oskf1/k); sprintf('%0.6f',RMSE_Oskf1)
disp('RMSE KF kecepatan translasi ke atas');
RMSE_kf2=sgt(MSE_kf2/k); sprintf('%0.6f',RMSE_kf2)
disp('RMSE OSKF kecepatan translasi ke atas');
RMSE_Oskf2=sgt(MSE_Oskf2/k); sprintf('%0.6f',RMSE_Oskf2)
disp('RMSE KF kecepatan sudut pitch');
RMSE_kf3=sgt(MSE_kf3/k); sprintf('%0.6f',RMSE_kf3)
disp('RMSE OSKF kecepatan sudut pitch');
RMSE_Oskf3=sgt(MSE_Oskf3/k); sprintf('%0.6f',RMSE_Oskf3)
disp('RMSE KF sudut pitch');
RMSE_kf4=sgt(MSE_kf4/k); sprintf('%0.6f',RMSE_kf4)
disp('RMSE OSKF sudut pitch');
RMSE_Oskf4=sgt(MSE_Oskf4/k); sprintf('%0.6f',RMSE_Oskf4)

figure(1);
plot(1:k+1,wra(1,:), 'b', 1:k+1,wfoor_kf(1,:), 'm', 1:k+1,wcoor_kf(1,:), 'g')
xlabel('Iterasi ke-')
ylabel('u')
grid on;
legend('Real', 'KF', 'Smoothing');
figure(2);
plot(1:k+1,wra(2,:), 'b', 1:k+1,wfoor_kf(2,:), 'm', 1:k+1,wcoor_kf(2,:), 'g')

```

vi)

```

xlabel('Iterasi ke-')
ylabel('w')
grid on;
legend('Real', 'KF', 'Smoothing');
figure(3);
plot(1:k+1, wre(3,:), 'b', 1:k+1, wfor_kf(3,:), 'r', 1:k+1, wcor_kf(3,:), 'g')
xlabel('Iterasi ke-')
ylabel('q')
grid on;
legend('Real', 'KF', 'Smoothing');
figure(4);
plot(1:k+1, wre(4,:), 'b', 1:k+1, wfor_kf(4,:), 'r', 1:k+1, wcor_kf(4,:), 'g')
xlabel('Iterasi ke-')
ylabel('theta')
grid on;
legend('Real', 'KF', 'Smoothing');

```

### 3. Program untuk estimasi *optimal smoothing* Fuzzy Kalman Filter

i)

```

clear all;
clc;

u0=1.5; w0=0.75; q0=0.1; theta0=0.05;
% parameter-parameter
k = 100; daltat=0.01;
U0=27.78; Xu=0.0002; Xw=0.039; g=9.8; Xda=0.44; Xdt=3.434*10^(-6);
zu=-0.07; zw=-0.317; Zda=-5.46; Zdt=-1.5*10^(-7); Mu=8.8*10^(-3);
Mw=-0.002873; Mg=-0.439; Mwdot=-0.0004; Mda=-1.1578; Mdt=6.701*10^(-7);
da=0.1; dt=0.1;
% Model sistem
Q=[1 0 0 1 0 0 0 1 0 0 0 1]; uk=[da;dt];
Q=[0.01 0 0 0 0.01*pi/180 0 0 0 0.0001*pi/180];
% Model pengukuran
H=[0 1 0 0 0 0 1 0]; R=[0.01 0 0 0 0.01*pi/180];
% Inisialisasi
x0=[u0;w0;q0;theta0];
u_kf=u0; w_kf=w0; q_kf=q0; theta_kf=theta0;
xf0x_kf(:,1)=x0; x0x_kf(:,1)=x0; xbp0x_kf(:,1,k+1)=[0;0;0;0];
P0=[0.3 0 0 0 0 0.3 0 0 0 0 0.3*pi/180 0 0 0 0 0.3*pi/180];
Ff0x_kf(:,1)=P0; Fbp0x_kf(:,1,k+1)=zeros(4,4);

error_kf1=0; error_0skf1=0; error_kf2=0; error_0skf2=0;
error_kf3=0; error_0skf3=0; error_kf4=0; error_0skf4=0;
MSE_kf1=0; MSE_0skf1=0; MSE_kf2=0; MSE_0skf2=0; MSE_kf3=0; MSE_0skf3=0;
MSE_kf4=0; MSE_0skf4=0;

% sistem real
xre(:,1)=x0;xre(:,2)=[1.4820;0.7573;0.0964;0.0514];xre(:,3)=[1.5243;0.82
39;0.0947;0.0523];
xre(:,4)=[1.5979;0.9213;0.0946;0.0530];xre(:,5)=[1.6544;1.0014;0.0925;0.05
45];xre(:,6)=[1.6328;1.0027;0.0916;0.0554];
xre(:,7)=[1.5986;0.9911;0.0891;0.0560];xre(:,8)=[1.6294;1.0439;0.0878;0.05
70];xre(:,9)=[1.5905;1.0266;0.0838;0.0582];
xre(:,10)=[1.6170;1.0735;0.0817;0.0589];xre(:,11)=[1.6466;1.1230;0.0835;0.
0602];xre(:,12)=[1.6349;1.1507;0.0841;0.0622];
xre(:,13)=[1.6405;1.1379;0.0822;0.0627];xre(:,14)=[1.6372;1.1747;0.0813;0.
0635];xre(:,15)=[1.6476;1.2049;0.0799;0.0647];
xre(:,16)=[1.6362;1.2129;0.0776;0.0660];xre(:,17)=[1.6761;1.2716;0.0783;0.
0670];xre(:,18)=[1.6511;1.2757;0.0765;0.0677];
xre(:,19)=[1.6507;1.2838;0.0728;0.0686];xre(:,20)=[1.6109;1.2615;0.0721;0.
0688];xre(:,21)=[1.6415;1.3094;0.0697;0.0698];
xre(:,22)=[1.6092;1.2936;0.0672;0.0707];xre(:,23)=[1.6416;1.3418;0.0647;0.
0708];xre(:,24)=[1.6202;1.3255;0.0640;0.0710];
xre(:,25)=[1.6349;1.3651;0.0625;0.0723];xre(:,26)=[1.6664;1.4110;0.0603;0.
0724];xre(:,27)=[1.6780;1.4363;0.0596;0.0731];
xre(:,28)=[1.6913;1.4621;0.0569;0.0739];xre(:,29)=[1.6718;1.4561;0.0553;0.
0751];xre(:,30)=[1.6455;1.4421;0.0532;0.0758];
xre(:,31)=[1.5984;1.4067;0.0525;0.0771];xre(:,32)=[1.6384;1.4584;0.0510;0.
0777];xre(:,33)=[1.6087;1.4398;0.0513;0.0785];
xre(:,34)=[1.5746;1.4172;0.0484;0.0786];xre(:,35)=[1.5718;1.4250;0.0474;0.
0784];xre(:,36)=[1.5839;1.4524;0.0439;0.0790];
xre(:,37)=[1.5708;1.4422;0.0420;0.0790];xre(:,38)=[1.5639;1.4449;0.0416;0.
0801];xre(:,39)=[1.5678;1.4579;0.0395;0.0808];
xre(:,40)=[1.5459;1.4434;0.0363;0.0815];xre(:,41)=[1.5256;1.4302;0.0353;0.
0812];xre(:,42)=[1.5349;1.4463;0.0363;0.0806];

```



ii)

```

xme(z,z,43)-[1.5079;1.4264;0.0322;0.0807];xme(z,z,44)-[1.4621;1.3864;0.0312;0.
0811];xme(z,z,45)-[1.4544;1.3845;0.0302;0.0816];
xme(z,z,46)-[1.4044;1.3450;0.0303;0.0811];xme(z,z,47)-[1.3815;1.3227;0.0300;0.
0813];xme(z,z,48)-[1.3454;1.2923;0.0306;0.0820];
xme(z,z,49)-[1.3228;1.2757;0.0294;0.0821];xme(z,z,50)-[1.2787;1.2374;0.0281;0.
0822];xme(z,z,51)-[1.2662;1.2305;0.0275;0.0819];
xme(z,z,52)-[1.3000;1.2696;0.0268;0.0815];xme(z,z,53)-[1.3356;1.3302;0.0243;0.
0821];xme(z,z,54)-[1.3634;1.3420;0.0255;0.0823];
xme(z,z,55)-[1.2753;1.2582;0.0244;0.0823];xme(z,z,56)-[1.2506;1.2379;0.0224;0.
0832];xme(z,z,57)-[1.2187;1.2097;0.0220;0.0841];
xme(z,z,58)-[1.2225;1.2175;0.0219;0.0845];xme(z,z,59)-[1.1990;1.1979;0.0217;0.
0851];xme(z,z,60)-[1.1681;1.1709;0.0214;0.0845];
xme(z,z,61)-[1.1241;1.1308;0.0197;0.0849];xme(z,z,62)-[1.1056;1.1158;0.0185;0.
0852];xme(z,z,63)-[1.1308;1.1644;0.0166;0.0864];
xme(z,z,64)-[1.1759;1.1922;0.0154;0.0868];xme(z,z,65)-[1.1582;1.1768;0.0133;0.
0869];xme(z,z,66)-[1.1924;1.2127;0.0119;0.0873];
xme(z,z,67)-[1.1967;1.2183;0.0128;0.0871];xme(z,z,68)-[1.2187;1.2418;0.0121;0.
0869];xme(z,z,69)-[1.2038;1.2281;0.0124;0.0860];
xme(z,z,70)-[1.1925;1.2179;0.0110;0.0866];xme(z,z,71)-[1.2080;1.2344;0.0100;0.
0865];xme(z,z,72)-[1.1966;1.2235;0.0097;0.0866];
xme(z,z,73)-[1.1695;1.1970;0.0098;0.0873];xme(z,z,74)-[1.1620;1.1901;0.0097;0.
0878];xme(z,z,75)-[1.1677;1.1967;0.0093;0.0880];
xme(z,z,76)-[1.1308;1.1604;0.0080;0.0881];xme(z,z,77)-[1.0838;1.1139;0.0046;0.
0883];xme(z,z,78)-[1.0687;1.0984;0.0022;0.0877];
xme(z,z,79)-[1.0882;1.1167;0.0001;0.0879];xme(z,z,80)-[1.1348;1.1615;-
0.0018;0.0883];xme(z,z,81)-[1.1565;1.1807;-0.0019;0.0881];
xme(z,z,82)-[1.1128;1.1344;-0.0027;0.0885];xme(z,z,83)-[1.1497;1.1687;-
0.0038;0.0887];xme(z,z,84)-[1.1197;1.1355;-0.0052;0.0890];
xme(z,z,85)-[1.1246;1.1371;-0.0074;0.0886];xme(z,z,86)-[1.1208;1.1292;-
0.0081;0.0878];xme(z,z,87)-[1.1196;1.1237;-0.0084;0.0873];
xme(z,z,88)-[1.0981;1.0979;-0.0075;0.0876];xme(z,z,89)-[1.0666;1.0622;-
0.0068;0.0876];xme(z,z,90)-[1.0488;1.0407;-0.0068;0.0877];
xme(z,z,91)-[1.0913;1.0797;-0.0078;0.0877];xme(z,z,92)-[1.0989;1.0832;-
0.0079;0.0871];xme(z,z,93)-[1.1180;1.0982;-0.0079;0.0870];
xme(z,z,94)-[1.0732;1.0490;-0.0084;0.0872];xme(z,z,95)-[1.0568;1.0284;-
0.0083;0.0863];xme(z,z,96)-[1.0664;1.0339;-0.0107;0.0853];
xme(z,z,97)-[1.0319;0.9944;-0.0105;0.0858];xme(z,z,98)-[1.0689;1.0267;-
0.0118;0.0847];xme(z,z,99)-[1.0447;0.9972;-0.0094;0.0843];
xme(z,z,100)-[1.0852;1.0333;-0.0083;0.0844];xme(z,z,101)-[1.0634;1.0073;-
0.0088;0.0843];

xme(z,z,1)-[0.7988;0.0975];xme(z,z,2)-[0.8200;0.0950];xme(z,z,3)-[0.9287;0.093
0];
xme(z,z,4)-[1.0447;0.0927];xme(z,z,5)-[1.0439;0.0914];xme(z,z,6)-[0.9857;0.089
5];
xme(z,z,7)-[1.0313;0.0876];xme(z,z,8)-[1.0397;0.0827];xme(z,z,9)-[1.1113;0.082
7];
xme(z,z,10)-[1.1047;0.0815];xme(z,z,11)-[1.1798;0.0833];xme(z,z,12)-[1.2327;0.
0812];
xme(z,z,13)-[1.1289;0.0783];xme(z,z,14)-[1.1849;0.0792];xme(z,z,15)-[1.1908;0.
0783];
xme(z,z,16)-[1.2987;0.0783];xme(z,z,17)-[1.3094;0.0762];xme(z,z,18)-[1.2988;0.
0731];
xme(z,z,19)-[1.3030;0.0732];xme(z,z,20)-[1.3272;0.0709];xme(z,z,21)-[1.3312;0.
0660];

```

iii)

```

xre(:,22)=[1.3624;0.0664];xre(:,23)=[1.3558;0.0649];xre(:,24)=[1.3510;0.
0630];
xre(:,25)=[1.4043;0.0592];xre(:,26)=[1.4497;0.0596];xre(:,27)=[1.4828;0.
0569];
xre(:,28)=[1.4657;0.0553];xre(:,29)=[1.4746;0.0511];xre(:,30)=[1.3951;0.
0526];
xre(:,31)=[1.4456;0.0514];xre(:,32)=[1.4312;0.0522];xre(:,33)=[1.4027;0.
0505];
xre(:,34)=[1.4432;0.0471];xre(:,35)=[1.4713;0.0450];xre(:,36)=[1.4291;0.
0405];
xre(:,37)=[1.4460;0.0423];xre(:,38)=[1.4809;0.0368];xre(:,39)=[1.4509;0.
0369];
xre(:,40)=[1.4319;0.0351];xre(:,41)=[1.4249;0.0393];xre(:,42)=[1.4497;0.
0318];
xre(:,43)=[1.3673;0.0309];xre(:,44)=[1.3578;0.0303];xre(:,45)=[1.3167;0.
0271];
xre(:,46)=[1.3531;0.0302];xre(:,47)=[1.2840;0.0290];xre(:,48)=[1.3312;0.
0300];
xre(:,49)=[1.3167;0.0295];xre(:,50)=[1.2619;0.0297];xre(:,51)=[1.2621;0.
0254];
xre(:,52)=[1.3693;0.0234];xre(:,53)=[1.3204;0.0240];xre(:,54)=[1.2430;0.
0247];
xre(:,55)=[1.2508;0.0221];xre(:,56)=[1.2955;0.0222];xre(:,57)=[1.2444;0.
0225];
xre(:,58)=[1.2006;0.0210];xre(:,59)=[1.1743;0.0197];xre(:,60)=[1.1119;0.
0209];
xre(:,61)=[1.1488;0.0201];xre(:,62)=[1.1514;0.0165];xre(:,63)=[1.1987;0.
0155];
xre(:,64)=[1.1476;0.0130];xre(:,65)=[1.2558;0.0102];xre(:,66)=[1.2425;0.
0112];
xre(:,67)=[1.2434;0.0118];xre(:,68)=[1.2327;0.0107];xre(:,69)=[1.2208;0.
0097];
xre(:,70)=[1.2606;0.0104];xre(:,71)=[1.2137;0.0080];xre(:,72)=[1.1705;0.
0119];
xre(:,73)=[1.2103;0.0070];xre(:,74)=[1.1619;0.0111];xre(:,75)=[1.1873;0.
0080];
xre(:,76)=[1.1200;0.0050];xre(:,77)=[1.0976;0.0022];xre(:,78)=[1.0826;0.
0007];
xre(:,79)=[1.1672;-0.0050];xre(:,80)=[1.1787;-
0.0018];xre(:,81)=[1.0924;-0.0027];
xre(:,82)=[1.1804;-0.0044];xre(:,83)=[1.2058;-
0.0049];xre(:,84)=[1.1552;-0.0071];
xre(:,85)=[1.1603;-0.0097];xre(:,86)=[1.1402;-
0.0098];xre(:,87)=[1.1162;-0.0048];
xre(:,88)=[1.0525;-0.0062];xre(:,89)=[1.0214;-
0.0081];xre(:,90)=[1.0912;-0.0083];
xre(:,91)=[1.1275;-0.0083];xre(:,92)=[1.1302;-
0.0067];xre(:,93)=[1.0780;-0.0076];
xre(:,94)=[0.9770;-0.0078];xre(:,95)=[1.0383;-
0.0104];xre(:,96)=[0.9448;-0.0135];
xre(:,97)=[1.0813;-0.0126];xre(:,98)=[0.9961;-
0.0109];xre(:,99)=[1.0282;-0.0083];
xre(:,100)=[0.9713;-0.0083];

for i=1:k
    u1=0.9*u_fkf; u2=1.1*u_fkf; w1=0.9*w_fkf; w2=1.1*w_fkf;

```

iv)

```

q1=0.9*q_fkf; q2=1.1*q_fkf;

miu_u1=(u_fkf-u1)/(u2-u1); miu_u2=(u2-u_fkf)/(u2-u1);
miu_w1=(w_fkf-w1)/(w2-w1); miu_w2=(w2-w_fkf)/(w2-w1);
miu_q1=(q_fkf-q1)/(q2-q1); miu_q2=(q2-q_fkf)/(q2-q1);

u_min=u1*miu_u1; u_max=u2*miu_u2; w_min=w1*miu_w1;
w_max=w2*miu_w2; q_min=q1*miu_q1; q_max=q2*miu_q2;

B_fkf=[Xda*deltat Xdt*deltat;Zda*deltat Zdt*deltat;Mda*deltat Mdt*deltat;0
0];
A_1(:,:,1)-(1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat;(2u+q_min)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_min)*deltat 0;0 0 deltat 1];
A_2(:,:,1)-(1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat;(2u+q_max)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_min)*deltat 0;0 0 deltat 1];
A_3(:,:,1)-(1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat;(2u+q_min)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_min)*deltat 0;0 0 deltat 1];
A_4(:,:,1)-(1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat;(2u+q_max)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_min)*deltat 0;0 0 deltat 1];
A_5(:,:,1)-(1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat;(2u+q_min)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_max)*deltat 0;0 0 deltat 1];
A_6(:,:,1)-(1+Xu*deltat Xw*deltat -w_min*deltat -
g*deltat;(2u+q_max)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_max)*deltat 0;0 0 deltat 1];
A_7(:,:,1)-(1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat;(2u+q_min)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_max)*deltat 0;0 0 deltat 1];
A_8(:,:,1)-(1+Xu*deltat Xw*deltat -w_max*deltat -
g*deltat;(2u+q_max)*deltat 1+2w*deltat U0*deltat 0;Mu*deltat Mw*deltat
1+(Mq+Mwdot*u_max)*deltat 0;0 0 deltat 1];
% Optimal Smoothing Kalman Filter
% Forward filter
% Time Update

Ffpre_1(:,:,1+1)-(A_1(:,:,1)*Ffcor_1(:,:,1)*t*transpose(A_1(:,:,1)))+(G*C*t*transp
ose(G));
Ffpre_2(:,:,1+1)-(A_2(:,:,1)*Ffcor_2(:,:,1)*t*transpose(A_2(:,:,1)))+(G*C*t*transp
ose(G));
Ffpre_3(:,:,1+1)-(A_3(:,:,1)*Ffcor_3(:,:,1)*t*transpose(A_3(:,:,1)))+(G*C*t*transp
ose(G));
Ffpre_4(:,:,1+1)-(A_4(:,:,1)*Ffcor_4(:,:,1)*t*transpose(A_4(:,:,1)))+(G*C*t*transp
ose(G));
Ffpre_5(:,:,1+1)-(A_5(:,:,1)*Ffcor_5(:,:,1)*t*transpose(A_5(:,:,1)))+(G*C*t*transp
ose(G));
Ffpre_6(:,:,1+1)-(A_6(:,:,1)*Ffcor_6(:,:,1)*t*transpose(A_6(:,:,1)))+(G*C*t*transp
ose(G));

```

v)

```

Pfpres_7(:,:,i+1)-(A_7(:,:,i)*Pfcors_7(:,:,i)*transpose(A_7(:,:,i)))+(G*Q*transpose(G));

Pfpres_8(:,:,i+1)-(A_8(:,:,i)*Pfcors_8(:,:,i)*transpose(A_8(:,:,i)))+(G*Q*transpose(G));

xfps_1(:,:,i+1)-A_1(:,:,i)*xpcors_1(:,:,i)+B_fxk*uk;
xfps_2(:,:,i+1)-A_2(:,:,i)*xpcors_2(:,:,i)+B_fxk*uk;
xfps_3(:,:,i+1)-A_3(:,:,i)*xpcors_3(:,:,i)+B_fxk*uk;
xfps_4(:,:,i+1)-A_4(:,:,i)*xpcors_4(:,:,i)+B_fxk*uk;
xfps_5(:,:,i+1)-A_5(:,:,i)*xpcors_5(:,:,i)+B_fxk*uk;
xfps_6(:,:,i+1)-A_6(:,:,i)*xpcors_6(:,:,i)+B_fxk*uk;
xfps_7(:,:,i+1)-A_7(:,:,i)*xpcors_7(:,:,i)+B_fxk*uk;
xfps_8(:,:,i+1)-A_8(:,:,i)*xpcors_8(:,:,i)+B_fxk*uk;

% Measurement Update

Kf_1(:,:,i+1)-Pfpres_1(:,:,i+1)*transpose(R)*inv(H*Pfpres_1(:,:,i+1)*transpose(H)+R);

Kf_2(:,:,i+1)-Pfpres_2(:,:,i+1)*transpose(R)*inv(H*Pfpres_2(:,:,i+1)*transpose(H)+R);

Kf_3(:,:,i+1)-Pfpres_3(:,:,i+1)*transpose(R)*inv(H*Pfpres_3(:,:,i+1)*transpose(H)+R);

Kf_4(:,:,i+1)-Pfpres_4(:,:,i+1)*transpose(R)*inv(H*Pfpres_4(:,:,i+1)*transpose(H)+R);

Kf_5(:,:,i+1)-Pfpres_5(:,:,i+1)*transpose(R)*inv(H*Pfpres_5(:,:,i+1)*transpose(H)+R);

Kf_6(:,:,i+1)-Pfpres_6(:,:,i+1)*transpose(R)*inv(H*Pfpres_6(:,:,i+1)*transpose(H)+R);

Kf_7(:,:,i+1)-Pfpres_7(:,:,i+1)*transpose(R)*inv(H*Pfpres_7(:,:,i+1)*transpose(H)+R);

Kf_8(:,:,i+1)-Pfpres_8(:,:,i+1)*transpose(R)*inv(H*Pfpres_8(:,:,i+1)*transpose(H)+R);

Pfcors_1(:,:,i+1)-(eye(4)-Kf_1(:,:,i+1)*H)*Pfpres_1(:,:,i+1);
Pfcors_2(:,:,i+1)-(eye(4)-Kf_2(:,:,i+1)*H)*Pfpres_2(:,:,i+1);
Pfcors_3(:,:,i+1)-(eye(4)-Kf_3(:,:,i+1)*H)*Pfpres_3(:,:,i+1);
Pfcors_4(:,:,i+1)-(eye(4)-Kf_4(:,:,i+1)*H)*Pfpres_4(:,:,i+1);
Pfcors_5(:,:,i+1)-(eye(4)-Kf_5(:,:,i+1)*H)*Pfpres_5(:,:,i+1);
Pfcors_6(:,:,i+1)-(eye(4)-Kf_6(:,:,i+1)*H)*Pfpres_6(:,:,i+1);
Pfcors_7(:,:,i+1)-(eye(4)-Kf_7(:,:,i+1)*H)*Pfpres_7(:,:,i+1);
Pfcors_8(:,:,i+1)-(eye(4)-Kf_8(:,:,i+1)*H)*Pfpres_8(:,:,i+1);

xpcors_1(:,:,i+1)-xfps_1(:,:,i+1)+Kf_1(:,:,i+1)*(zss(:,:,i)-H*xfps_1(:,:,i+1));
xpcors_2(:,:,i+1)-xfps_2(:,:,i+1)+Kf_2(:,:,i+1)*(zss(:,:,i)-H*xfps_2(:,:,i+1));

```

vi)

```

        xfcor_3(:, :, i+1) - xfpres_3(:, :, i+1) + Rf_3(:, :, i+1) * (zre(:, :, i) -
        H*xfpres_3(:, :, i+1));
        xfcor_4(:, :, i+1) - xfpres_4(:, :, i+1) + Rf_4(:, :, i+1) * (zre(:, :, i) -
        H*xfpres_4(:, :, i+1));
        xfcor_5(:, :, i+1) - xfpres_5(:, :, i+1) + Rf_5(:, :, i+1) * (zre(:, :, i) -
        H*xfpres_5(:, :, i+1));
        xfcor_6(:, :, i+1) - xfpres_6(:, :, i+1) + Rf_6(:, :, i+1) * (zre(:, :, i) -
        H*xfpres_6(:, :, i+1));
        xfcor_7(:, :, i+1) - xfpres_7(:, :, i+1) + Rf_7(:, :, i+1) * (zre(:, :, i) -
        H*xfpres_7(:, :, i+1));
        xfcor_8(:, :, i+1) - xfpres_8(:, :, i+1) + Rf_8(:, :, i+1) * (zre(:, :, i) -
        H*xfpres_8(:, :, i+1));

        % Proses Defensifikasi
        W_1(:, :, i) - miu_u1*miu_w1*miu_q1; W_2(:, :, i) - miu_u1*miu_w1*miu_q2;
        W_3(:, :, i) - miu_u1*miu_w2*miu_q1; W_4(:, :, i) - miu_u1*miu_w2*miu_q2;
        W_5(:, :, i) - miu_u2*miu_w1*miu_q1; W_6(:, :, i) - miu_u2*miu_w1*miu_q2;
        W_7(:, :, i) - miu_u2*miu_w2*miu_q1; W_8(:, :, i) - miu_u2*miu_w2*miu_q2;

        xfcor_fkf(:, :, i+1) - (W_1(:, :, i)*xfcor_1(:, :, i+1) + W_2(:, :, i)*xfcor_2(:, :, i+1) + W_
        3(:, :, i)*xfcor_3(:, :, i+1) + W_4(:, :, i)*xfcor_4(:, :, i+1) + W_5(:, :, i)*xfcor_5(:, :, i
        +1) + W_6(:, :, i)*xfcor_6(:, :, i+1) + W_7(:, :, i)*xfcor_7(:, :, i+1) + W_8(:, :, i)*xfcor_8
        (:, :, i+1)) / (W_1(:, :, i) + W_2(:, :, i) + W_3(:, :, i) + W_4(:, :, i) + W_5(:, :, i) + W_6(:, :, i) +
        W_7(:, :, i) + W_8(:, :, i));

        u_fkf - xfcor_fkf(1, :, i+1); w_fkf - xfcor_fkf(2, :, i+1);
        Q_fkf - xfcor_fkf(3, :, i+1); theta_fkf - xfcor_fkf(4, :, i+1);

and
for i=1:k
    % Backward filter
    % Measurement Update
    Fbcor_1(:, :, k-i+2) - Fbpres_1(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_2(:, :, k-i+2) - Fbpres_2(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_3(:, :, k-i+2) - Fbpres_3(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_4(:, :, k-i+2) - Fbpres_4(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_5(:, :, k-i+2) - Fbpres_5(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_6(:, :, k-i+2) - Fbpres_6(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_7(:, :, k-i+2) - Fbpres_7(:, :, k-i+2) + transpose(H) * inv(R) * H;
    Fbcor_8(:, :, k-i+2) - Fbpres_8(:, :, k-i+2) + transpose(H) * inv(R) * H;

    xbcor_1(:, :, k-i+2) - xbpres_1(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);
    xbcor_2(:, :, k-i+2) - xbpres_2(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);
    xbcor_3(:, :, k-i+2) - xbpres_3(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);
    xbcor_4(:, :, k-i+2) - xbpres_4(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);
    xbcor_5(:, :, k-i+2) - xbpres_5(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);
    xbcor_6(:, :, k-i+2) - xbpres_6(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);
    xbcor_7(:, :, k-i+2) - xbpres_7(:, :, k-i+2) + transpose(H) * inv(R) * zre(:, :, k-
    i+1);

```

vii)

```

xhcor_8(:, :, k-i+2) - xhpr_8(:, :, k-i+2) + transpose(R) * inv(R) * xre(:, :, k-
1+1);

% Time Update
Kb_1(:, :, k-i+1) - Fhcor_1(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_1(:, :, k-
1+2) * G + inv(Q));
Kb_2(:, :, k-i+1) - Fhcor_2(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_2(:, :, k-
1+2) * G + inv(Q));
Kb_3(:, :, k-i+1) - Fhcor_3(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_3(:, :, k-
1+2) * G + inv(Q));
Kb_4(:, :, k-i+1) - Fhcor_4(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_4(:, :, k-
1+2) * G + inv(Q));
Kb_5(:, :, k-i+1) - Fhcor_5(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_5(:, :, k-
1+2) * G + inv(Q));
Kb_6(:, :, k-i+1) - Fhcor_6(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_6(:, :, k-
1+2) * G + inv(Q));
Kb_7(:, :, k-i+1) - Fhcor_7(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_7(:, :, k-
1+2) * G + inv(Q));
Kb_8(:, :, k-i+1) - Fhcor_8(:, :, k-i+2) * G * inv(transpose(G) * Fhcor_8(:, :, k-
1+2) * G + inv(Q));

Fhpr_1(:, :, k-i+1) - transpose(R_1(:, :, k-i+1)) * (eye(4) - Kb_1(:, :, k-
1+1) * transpose(G)) * Fhcor_1(:, :, k-i+2) * R_1(:, :, k-i+1);
Fhpr_2(:, :, k-i+1) - transpose(R_2(:, :, k-i+1)) * (eye(4) - Kb_2(:, :, k-
1+1) * transpose(G)) * Fhcor_2(:, :, k-i+2) * R_2(:, :, k-i+1);
Fhpr_3(:, :, k-i+1) - transpose(R_3(:, :, k-i+1)) * (eye(4) - Kb_3(:, :, k-
1+1) * transpose(G)) * Fhcor_3(:, :, k-i+2) * R_3(:, :, k-i+1);
Fhpr_4(:, :, k-i+1) - transpose(R_4(:, :, k-i+1)) * (eye(4) - Kb_4(:, :, k-
1+1) * transpose(G)) * Fhcor_4(:, :, k-i+2) * R_4(:, :, k-i+1);
Fhpr_5(:, :, k-i+1) - transpose(R_5(:, :, k-i+1)) * (eye(4) - Kb_5(:, :, k-
1+1) * transpose(G)) * Fhcor_5(:, :, k-i+2) * R_5(:, :, k-i+1);
Fhpr_6(:, :, k-i+1) - transpose(R_6(:, :, k-i+1)) * (eye(4) - Kb_6(:, :, k-
1+1) * transpose(G)) * Fhcor_6(:, :, k-i+2) * R_6(:, :, k-i+1);
Fhpr_7(:, :, k-i+1) - transpose(R_7(:, :, k-i+1)) * (eye(4) - Kb_7(:, :, k-
1+1) * transpose(G)) * Fhcor_7(:, :, k-i+2) * R_7(:, :, k-i+1);
Fhpr_8(:, :, k-i+1) - transpose(R_8(:, :, k-i+1)) * (eye(4) - Kb_8(:, :, k-
1+1) * transpose(G)) * Fhcor_8(:, :, k-i+2) * R_8(:, :, k-i+1);

xhpr_1(:, :, k-i+1) - transpose(R_1(:, :, k-i+1)) * (eye(4) - Kb_1(:, :, k-
1+1) * transpose(G)) * (xhcor_1(:, :, k-i+2) - Fhcor_1(:, :, k-i+2) * B * Kf * uk);
xhpr_2(:, :, k-i+1) - transpose(R_2(:, :, k-i+1)) * (eye(4) - Kb_2(:, :, k-
1+1) * transpose(G)) * (xhcor_2(:, :, k-i+2) - Fhcor_2(:, :, k-i+2) * B * Kf * uk);
xhpr_3(:, :, k-i+1) - transpose(R_3(:, :, k-i+1)) * (eye(4) - Kb_3(:, :, k-
1+1) * transpose(G)) * (xhcor_3(:, :, k-i+2) - Fhcor_3(:, :, k-i+2) * B * Kf * uk);
xhpr_4(:, :, k-i+1) - transpose(R_4(:, :, k-i+1)) * (eye(4) - Kb_4(:, :, k-
1+1) * transpose(G)) * (xhcor_4(:, :, k-i+2) - Fhcor_4(:, :, k-i+2) * B * Kf * uk);
xhpr_5(:, :, k-i+1) - transpose(R_5(:, :, k-i+1)) * (eye(4) - Kb_5(:, :, k-
1+1) * transpose(G)) * (xhcor_5(:, :, k-i+2) - Fhcor_5(:, :, k-i+2) * B * Kf * uk);
xhpr_6(:, :, k-i+1) - transpose(R_6(:, :, k-i+1)) * (eye(4) - Kb_6(:, :, k-
1+1) * transpose(G)) * (xhcor_6(:, :, k-i+2) - Fhcor_6(:, :, k-i+2) * B * Kf * uk);
xhpr_7(:, :, k-i+1) - transpose(R_7(:, :, k-i+1)) * (eye(4) - Kb_7(:, :, k-
1+1) * transpose(G)) * (xhcor_7(:, :, k-i+2) - Fhcor_7(:, :, k-i+2) * B * Kf * uk);
xhpr_8(:, :, k-i+1) - transpose(R_8(:, :, k-i+1)) * (eye(4) - Kb_8(:, :, k-
1+1) * transpose(G)) * (xhcor_8(:, :, k-i+2) - Fhcor_8(:, :, k-i+2) * B * Kf * uk);
end
for i=1:L

```

viii)

```

% Smoother
K_1(:,:,1+1)-Pfcov_1(:,:,1+1)*Pbpres_1(:,:,1+1)*inv(eye(4)+Pfcov_1(:,:,1+1)*Pbp
=a_1(:,:,1+1));

K_2(:,:,1+1)-Pfcov_2(:,:,1+1)*Pbpres_2(:,:,1+1)*inv(eye(4)+Pfcov_2(:,:,1+1)*Pbp
=a_2(:,:,1+1));

K_3(:,:,1+1)-Pfcov_3(:,:,1+1)*Pbpres_3(:,:,1+1)*inv(eye(4)+Pfcov_3(:,:,1+1)*Pbp
=a_3(:,:,1+1));

K_4(:,:,1+1)-Pfcov_4(:,:,1+1)*Pbpres_4(:,:,1+1)*inv(eye(4)+Pfcov_4(:,:,1+1)*Pbp
=a_4(:,:,1+1));

K_5(:,:,1+1)-Pfcov_5(:,:,1+1)*Pbpres_5(:,:,1+1)*inv(eye(4)+Pfcov_5(:,:,1+1)*Pbp
=a_5(:,:,1+1));

K_6(:,:,1+1)-Pfcov_6(:,:,1+1)*Pbpres_6(:,:,1+1)*inv(eye(4)+Pfcov_6(:,:,1+1)*Pbp
=a_6(:,:,1+1));

K_7(:,:,1+1)-Pfcov_7(:,:,1+1)*Pbpres_7(:,:,1+1)*inv(eye(4)+Pfcov_7(:,:,1+1)*Pbp
=a_7(:,:,1+1));

K_8(:,:,1+1)-Pfcov_8(:,:,1+1)*Pbpres_8(:,:,1+1)*inv(eye(4)+Pfcov_8(:,:,1+1)*Pbp
=a_8(:,:,1+1));

F_1(:,:,1+1)-(eye(4)-K_1(:,:,1+1))*Pfcov_1(:,:,1+1);
F_2(:,:,1+1)-(eye(4)-K_2(:,:,1+1))*Pfcov_2(:,:,1+1);
F_3(:,:,1+1)-(eye(4)-K_3(:,:,1+1))*Pfcov_3(:,:,1+1);
F_4(:,:,1+1)-(eye(4)-K_4(:,:,1+1))*Pfcov_4(:,:,1+1);
F_5(:,:,1+1)-(eye(4)-K_5(:,:,1+1))*Pfcov_5(:,:,1+1);
F_6(:,:,1+1)-(eye(4)-K_6(:,:,1+1))*Pfcov_6(:,:,1+1);
F_7(:,:,1+1)-(eye(4)-K_7(:,:,1+1))*Pfcov_7(:,:,1+1);
F_8(:,:,1+1)-(eye(4)-K_8(:,:,1+1))*Pfcov_8(:,:,1+1);

xcof_1(:,:,1+1)-(eye(4)-
K_1(:,:,1+1))*xcof_1(:,:,1+1)+F_1(:,:,1+1)*xbpres_1(:,:,1+1);
xcof_2(:,:,1+1)-(eye(4)-
K_2(:,:,1+1))*xcof_2(:,:,1+1)+F_2(:,:,1+1)*xbpres_2(:,:,1+1);
xcof_3(:,:,1+1)-(eye(4)-
K_3(:,:,1+1))*xcof_3(:,:,1+1)+F_3(:,:,1+1)*xbpres_3(:,:,1+1);
xcof_4(:,:,1+1)-(eye(4)-
K_4(:,:,1+1))*xcof_4(:,:,1+1)+F_4(:,:,1+1)*xbpres_4(:,:,1+1);
xcof_5(:,:,1+1)-(eye(4)-
K_5(:,:,1+1))*xcof_5(:,:,1+1)+F_5(:,:,1+1)*xbpres_5(:,:,1+1);
xcof_6(:,:,1+1)-(eye(4)-
K_6(:,:,1+1))*xcof_6(:,:,1+1)+F_6(:,:,1+1)*xbpres_6(:,:,1+1);
xcof_7(:,:,1+1)-(eye(4)-
K_7(:,:,1+1))*xcof_7(:,:,1+1)+F_7(:,:,1+1)*xbpres_7(:,:,1+1);
xcof_8(:,:,1+1)-(eye(4)-
K_8(:,:,1+1))*xcof_8(:,:,1+1)+F_8(:,:,1+1)*xbpres_8(:,:,1+1);

xcof_1xf(:,:,1+1)-(W_1(:,:,1)*xcof_1(:,:,1+1)+W_2(:,:,1)*xcof_2(:,:,1+1)+W_3(
(:,:,1)*xcof_3(:,:,1+1)+W_4(:,:,1)*xcof_4(:,:,1+1)+W_5(:,:,1)*xcof_5(:,:,1+1)+W_
6(:,:,1)*xcof_6(:,:,1+1)+W_7(:,:,1)*xcof_7(:,:,1+1)+W_8(:,:,1)*xcof_8(:,:,1+1)

```

ix)

```

) / (W_1(:, :, 1) + W_2(:, :, 1) + W_3(:, :, 1) + W_4(:, :, 1) + W_5(:, :, 1) + W_6(:, :, 1) + W_7(:, :, 1)
) + W_8(:, :, 1));
end
for i=1:k
    % Error kecepatan translasi ke depan
    error_fkf1=(error_fkf1 abs(xse(1, :, i+1)-wcor_fkf(1, :, i+1)));
    error_osfkf1=(error_osfkf1 abs(xse(1, :, i+1)-wcor_fkf(1, :, i+1)));
    % Error kecepatan translasi ke atas
    error_fkf2=(error_fkf2 abs(xse(2, :, i+1)-wcor_fkf(2, :, i+1)));
    error_osfkf2=(error_osfkf2 abs(xse(2, :, i+1)-wcor_fkf(2, :, i+1)));
    % Error kecepatan sudut pitch
    error_fkf3=(error_fkf3 abs(xse(3, :, i+1)-wcor_fkf(3, :, i+1)));
    error_osfkf3=(error_osfkf3 abs(xse(3, :, i+1)-wcor_fkf(3, :, i+1)));
    % Error sudut pitch
    error_fkf4=(error_fkf4 abs(xse(4, :, i+1)-wcor_fkf(4, :, i+1)));
    error_osfkf4=(error_osfkf4 abs(xse(4, :, i+1)-wcor_fkf(4, :, i+1)));
    % MSE kecepatan translasi ke depan
    MSE_fkf1-MSE_fkf1+(xse(1, :, i+1)-wcor_fkf(1, :, i+1))^2;
    MSE_osfkf1-MSE_osfkf1+(xse(1, :, i+1)-wcor_fkf(1, :, i+1))^2;
    % MSE kecepatan translasi ke atas
    MSE_fkf2-MSE_fkf2+(xse(2, :, i+1)-wcor_fkf(2, :, i+1))^2;
    MSE_osfkf2-MSE_osfkf2+(xse(2, :, i+1)-wcor_fkf(2, :, i+1))^2;
    % MSE kecepatan sudut pitch
    MSE_fkf3-MSE_fkf3+(xse(3, :, i+1)-wcor_fkf(3, :, i+1))^2;
    MSE_osfkf3-MSE_osfkf3+(xse(3, :, i+1)-wcor_fkf(3, :, i+1))^2;
    % MSE sudut pitch
    MSE_fkf4-MSE_fkf4+(xse(4, :, i+1)-wcor_fkf(4, :, i+1))^2;
    MSE_osfkf4-MSE_osfkf4+(xse(4, :, i+1)-wcor_fkf(4, :, i+1))^2;
end
diap('Error FKF kecepatan translasi ke depan');
sprintf('%0.6f', error_fkf1);
diap('Error OSFKF kecepatan translasi ke depan');
sprintf('%0.6f', error_osfkf1);
diap('Error FKF kecepatan translasi ke atas');
sprintf('%0.6f', error_fkf2);
diap('Error OSFKF kecepatan translasi ke atas');
sprintf('%0.6f', error_osfkf2);
diap('Error FKF kecepatan sudut pitch');
sprintf('%0.6f', error_fkf3);
diap('Error OSFKF kecepatan sudut pitch');
sprintf('%0.6f', error_osfkf3);
diap('Error FKF sudut pitch');
sprintf('%0.6f', error_fkf4);
diap('Error OSFKF sudut pitch');
sprintf('%0.6f', error_osfkf4);
diap('RMSE FKF kecepatan translasi ke depan');
RMSE_fkf1=sgt(MSE_fkf1/k); sprintf('%0.6f', RMSE_fkf1);
diap('RMSE OSFKF kecepatan translasi ke depan');
RMSE_osfkf1=sgt(MSE_osfkf1/k); sprintf('%0.6f', RMSE_osfkf1);
diap('RMSE FKF kecepatan translasi ke atas');
RMSE_fkf2=sgt(MSE_fkf2/k); sprintf('%0.6f', RMSE_fkf2);
diap('RMSE OSFKF kecepatan translasi ke atas');
RMSE_osfkf2=sgt(MSE_osfkf2/k); sprintf('%0.6f', RMSE_osfkf2);
diap('RMSE FKF kecepatan sudut pitch');
RMSE_fkf3=sgt(MSE_fkf3/k); sprintf('%0.6f', RMSE_fkf3);
diap('RMSE OSFKF kecepatan sudut pitch');
RMSE_osfkf3=sgt(MSE_osfkf3/k); sprintf('%0.6f', RMSE_osfkf3);

```



x)

```

disp('RMSE FKF sudut pitch');
RMSE_fkf4=sqrt(MSE_fkf4/k); sprintf('%0.6f',RMSE_fkf4)
disp('RMSE OSFKF sudut pitch');
RMSE_OSfkf4=sqrt(MSE_OSfkf4/k); sprintf('%0.6f',RMSE_OSfkf4)

figure(1);
plot(1:k+1,xre(1,:), 'b', 1:k+1,xfcor_fkf(1,:), 'r', 1:k+1,xcor_fkf(1,:), 'g')
title('Estimasi kecepatan translasi ke depan')
xlabel('Iterasi Ke-')
ylabel('u')
grid on;
legend('Real', 'FKF', 'Smoothing');
figure(2);
plot(1:k+1,xre(2,:), 'b', 1:k+1,xfcor_fkf(2,:), 'r', 1:k+1,xcor_fkf(2,:), 'g')
title('Estimasi kecepatan translasi ke atas')
xlabel('Iterasi Ke-')
ylabel('w')
grid on;
legend('Real', 'FKF', 'Smoothing');
figure(3);
plot(1:k+1,xre(3,:), 'b', 1:k+1,xfcor_fkf(3,:), 'r', 1:k+1,xcor_fkf(3,:), 'g')
title('Estimasi kecepatan sudut pitch')
xlabel('Iterasi Ke-')
ylabel('q')
grid on;
legend('Real', 'FKF', 'Smoothing');
figure(4);
plot(1:k+1,xre(4,:), 'b', 1:k+1,xfcor_fkf(4,:), 'r', 1:k+1,xcor_fkf(4,:), 'g')
title('Estimasi sudut pitch')
xlabel('Iterasi Ke-')
ylabel('theta')
grid on;
legend('Real', 'FKF', 'Smoothing');

```

*"Halaman ini sengaja dikosongkan."*

## BAB V

### PENUTUP

Pada bab ini, diberikan kesimpulan yang diperoleh dari Tugas Akhir ini serta saran untuk penelitian selanjutnya.

#### 5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat disimpulkan beberapa hal berikut:

1. Hasil estimasi menunjukkan bahwa setiap variabel gerak longitudinal pesawat terbang metode Fuzzy Kalman Filter lebih baik daripada metode Kalman Filter. Nilai RMSE juga menunjukkan bahwa error rata-rata dari metode Fuzzy Kalman Filter lebih kecil daripada metode Kalman Filter. Peningkatan error kecepatan translasi ke depan 62.4 %, kecepatan translasi ke atas 0.7 %, kecepatan sudut *pitch* 0.009 % dan sudut *pitch* 1.7 %.
2. Berdasarkan waktu komputasi menunjukkan bahwa metode Kalman Filter hanya membutuhkan waktu 0.1628s lebih baik daripada metode Fuzzy Kalman Filter yang membutuhkan waktu 0.2802 s.
3. Pada *optimal smoothing* Kalman Filter, hanya variabel kecepatan translasi ke atas, kecepatan sudut *pitch* dan sudut *pitch* yang dapat memberikan hasil estimasi lebih halus daripada Kalman Filter dengan peningkatan error yaitu kecepatan translasi ke atas 9.1 %, kecepatan sudut *pitch* 30.9 % dan sudut *pitch* 15.2 %. Pada variabel kecepatan translasi ke depan, hasil estimasi

Kalman Filter lebih baik dibandingkan pada saat *optimal smoothing*.

4. Pada *optimal smoothing* Fuzzy Kalman Filter, semua variabel dapat memberikan hasil estimasi yang lebih halus daripada Fuzzy Kalman Filter. Peningkatan error masing-masing yaitu kecepatan translasi ke depan 33.8 %, kecepatan translasi ke atas 8.3 %, kecepatan sudut *pitch* 31 % dan sudut *pitch* 14.4 %.

## 5.2 Saran

Pada Tugas Akhir ini, model gerak longitudinal pesawat terbang merupakan sistem model nonlinear. Oleh karena itu, untuk penelitian selanjutnya dapat digunakan dengan metode Extended Kalman Filter. Dengan melakukan kombinasi metode Logika Fuzzy dan Extended Kalman Filter.

## DAFTAR PUSTAKA

- [1] Kallapur, A. G. (2007). "UAV Linear and Nonlinear Estimation Using Eextended Kalman Filter". University of New South Wales.
- [2] Purwanto, E. B. (2003). "Pemodelan Sistem dan Analisis Kestabilan Dinamik Pesawat UAV". Jurnal Teknologi Dirgantara. Laporan Penelitian, LAPAN.
- [3] Salman, S. A. Sreenatha, A. G. Choi, J. T. (2006). "Attitude Dynamics Identification of Unmanned Aircraft Vehicle". Internasional Journal of Control, Automation and System.
- [4] Mahmuri, H. (2011). "Estimasi Perkembangan Sel Kanker Menggunakan Fuzzy Kalman Filter". Jurusan Matematika Institut Teknologi Sepuluh Nopember.
- [5] Sukandi, A. (2010). "Pengendalian Gerak Longitudinal Pesawat Terbang dengan Metode *Decoupling*". Jurusan Teknik Mesin Politeknik Negeri Jakarta.
- [6] Artono, E. (2010). "Pengembangan Sistem Pengendali Posisi Sudut dari Sirip Elevator pada Pesawat Tanpa Awak". Fakultas Teknik Universitas Indonesia.
- [7] Zimmermann, H. J. (1992). **Fuzzy Set Theory and Its Applications**. Second Revised Edition. United States.
- [8] Lewis, F. L. (1998). **Optimal Estimation with An Introduction to Stochastic Control Theory**.

School of Electrical Engineering Georgia Institute of Technology Atlanta. Georgia.

- [9] Han, L. R. (2004). "A Fuzzy-Kalman Filtering Strategy for State Estimation". A Thesis submitted to the College of Graduate Studies and Research in partial Fulfillment of the Requirements for the the Degree of Master of Science. In the Department of Mechanical Engineering University of Saskatchewan. Saskaton, Kanada.
- [10] McLean, D. (1990). **Automatic Flight Control Systems**. Prentice Hall International. UK.
- [11] Chen, G. Xie, Q. Shieh, L. S. (1997). "Fuzzy Kalman Filter". Journal Information of Information Sciences. No. 109, hal. 197-209.
- [12] Purnawan, H. (2015). "Desain Sistem Kendali linear Quadratic Regulator (LQR) untuk Kestabilan Terbang LSU-05". Tugas Akhir. Jurusan Matematika Institut Teknologi Sepuluh Nopember.
- [13] Muzaki, M.A.F. (2015). "Desain Kendali Sudut *Pitch* dan *Roll* pada LSU-05 dengan Metode *Sliding Mode Control (SMC)*". Tugas Akhir. Jurusan Matematika Institut Teknologi Sepuluh Nopember.
- [14] Ermayanti, Z. (2015). "Estimasi dan Kendali Posisi *Autonomous Underwater Vehicle* Berdasarkan Lintasan yang Ditentukan". Jurusan Matematika Institut Teknologi Sepuluh Nopember.

## **LAMPIRAN B**

### **Biodata Penulis**



Penulis bernama Resi Arumin Sani, yang biasanya dipanggil Resi. Penulis dilahirkan di Surabaya, 21 Desember 1993. Penulis merupakan putra dari pasangan Sulimin dan Nurul Ainiyah. Penulis menempuh pendidikan formal dimulai dari TK Dharma Wanita Jemundo (1998-2000), SDN Jemundo I (2000-2006), SMP Negeri 2 Taman Sidoarjo (2006-2009) dan SMA Wachid Hasyim 2 Taman YPM Sidoarjo (2009-2012). Penulis melanjutkan studi ke jenjang S1 di Jurusan Matematika ITS Surabaya pada tahun 2012 dengan NRP 1212 100 039. Penulis mengambil Bidang Minat Matematika Terapan. Selama kuliah, penulis memiliki pengalaman berorganisasi di KM ITS sebagai staf Departemen Pengabdian Masyarakat HIMATIKA ITS (2013-2014), staf Departemen Kaderisasi LDJ Ibnu Muqlah (2013-2014) dan Kepala Departemen Kaderisasi LDJ Ibnu Muqlah (2014-2015). Selain itu, penulis juga aktif dalam pelatihan kemahasiswaan, yaitu LKMM Pra-TD dan LKMM TD. Disamping itu, sejak semester V penulis terdaftar sebagai asisten dosen mata kuliah Kalkulus I dan Kalkulus II.

Informasi lebih lanjut mengenai Tugas Akhir ini dapat ditujukan ke penulis melalui email: [resi12@mhs.matematika.its.ac.id](mailto:resi12@mhs.matematika.its.ac.id)